A Basis for the Exploration of Hypermedia Systems: A Guided Path Facility

A thesis presented
in partial fulfillment of the requirements
for the degree of Master of Arts
at Massey University

Alan Stenhouse

1992

Abstract

This thesis examines the potential of a paths facility as an aid to navigating large hypermedia systems. The use of the navigational metaphor as applied to finding information is continued with the idea of following a path through information 'space'. This idea assumes that each node, or chunk of information, on the path can be considered a landmark that can be easily returned to when side-trips are taken off the path to explore the surrounding space.

The idea of a guided path assumes the re-use of a path, and also assumes that there is extra information available about the path. This meta-information is very important for providing information to help path-followers make better sense of the path, both in terms of content and context, but also in making more effective use of the nodes on the path and in navigating the variety of interface conventions seen in the test environment — HyperCard.

A small pilot study has been carried out using two groups of users performing a directed information-seeking task. One group used HyperCard's navigational facilities to find information in a group of stacks, while the other group used a guided path as a base on which to explore the same group of stacks. Both groups had a time limit, at the end of which they completed a number of questionnaires to indicate task completion, as well as providing a subjective evaluation of the facilities they used.

The guided path facility appears to be most effective for inexperienced users for a number of reasons. It presents a simplified view of the complex system — the information available has already been filtered and selected, and a simple and consistent navigational interface reduces the cognitive overheads associated with learning a variety of mechanisms present in different stacks.

An important feature of a path facility seems to be the provision of meta-information, especially scope information which can reduce the incidences of disorientation. Another feature is the provision of a history facility which provides a backtracking capability. It may also be used in the creation of paths using the length of visit as a criterion for node inclusion on a new path.

Acknowledgements

I gratefully acknowledge the advice and assistance of my supervisor, Professor Mark Apperley, without whose guidance this thesis may never have been completed. A constructive proof-reader was my father who corrected my bad grammar and was always keen to discuss the ideas contained within the thesis. Thanks are also due to the Brenden, Philip, and Colin for their assistance, and the rest of the Department for the use of their facilities. Despite all this help, I am sure that some mistakes will inevitably creep through. Any remaining errors are my sole responsibility.

Table of Contents

List of Figures

List of Tables

Introduction

Technological advances, as well as economic and environmental concerns, have resulted in the possibility of, and the demand for, electronic publishing. In order to make the published materials more effective for users (and more enticing to them), multiple media are being used to enhance the products. Hypertext-style systems are becoming more widely used in these products because they offer a great deal of flexibility — multiple ways of navigating a system mean that many different meanings can be taken from the same set of materials. Thus many users can, if so desired, access similar materials in different manners and in different sequences, and thus personalise their information paths, depending on their interests and preferred methods of interaction. The addition of multiple media has produced hypermedia systems that can link not only text to other text, but they can also link each medium to another. So we may have links from video to text, or audio to graphics. Thus the scope of these systems has increased greatly, as has their complexity.

Navigation Problems

This increase in complexity results in many problems, not only for the authors who must decide on what information should be linked to what, and in what format it should appear, but also for the readers of the finished documents who must first of all decipher what is linked and what isn't, and then decide on which links to follow. Without the cues that are used for navigating within traditional information sources such as books and libraries, the facilities provided in the systems and by the authors must be used to find the information required.

As the amount of information within these hypermedia systems increases, the problems that are faced in finding particular items of information increase as well. Following links from one chunk of information to another can rapidly result in a user becoming disoriented or lost. This means there is a need for mechanisms to reduce the occurrences of disorientation, and also for aids that can help a user to overcome the problem when it does occur.

In addition to the increase in complexity of the systems, another factor that can greatly affect the effective use of hypermedia systems is the amount of experience of the users. Experienced users can be expected to have fewer problems in utilising a system efficiently. Inexperienced or novice users would be expected to have more problems, not only in using the system controls, but also in understanding the structures within the system, or how the linked items relate to one another.

Every hypermedia system contains a variety of mechanisms to aid users in their information navigation tasks. Often these aids will be in the form of a graphical browser or map, a table of contents or overview, an index, a search facility, some sort of 'history' or go-back function, as well as the obvious follow-a-link mechanism. These facilities are reasonably effective for small- and medium-scale systems. When the size of the system increases, these facilities in turn become more difficult to use.

One method of reducing the system complexity is to provide some sort of filter that can simplify the view of the structures that need to be navigated. This filtering may be automatic, in the form of an intelligent agent, or it might be manual, in the form of a guided path. The resulting view contains information that is deemed relevant in some way to the information-seeker's needs.

However, for novice users, the act of specifying a filter, or knowing what to filter, is itself difficult without a good idea of the structure of the system or the subject area. So knowing what to filter out becomes very hard without this prior knowledge. Even if we assume that a filter is provided for a novice user, the cognitive overheads involved in actually trying to navigate the structure can detract from understanding the information contained in the system. A novice user may not understand the screen layout or the navigational controls.

A Path Mechanism

This means that additional methods of providing system simplification may be required. In addition to making the structure of the system easier to understand and to navigate, the actual interface may need to be simplified so that novice users can still navigate effectively with a reduced set of commands. This implies that the path can be navigated with a reduced number of controls, and as a path is basically linear then the primary controls are a step forward and a step back. Interface simplification also implies explaining screen layout conventions and navigational controls that are used in each card or stack on the path. So this means there is a requirement for some extra information supplied by the path author. This extra or meta-information can explain what's contained in the path, the interface conventions at each card on the path, noteworthy points, or it might provide supplementary information about the path topic. It might also be used by a path-follower to note down thoughts as they occur in the context that they occur. Thus a way to add or display this meta-information is needed in addition to the simplified navigational controls.

Thesis Proposal

This thesis proposes that a guided paths facility would be a useful addition to the navigational aids provided in hypermedia systems. It would be valuable to inexperienced users because it can present a simplified view of a large and complex hypermedia system. It also provides a simplified set of navigational controls that eases the cognitive load associated with understanding what different controls and conventions do in multiple stacks. In addition, a guided path actually 'guides' a path-follower to interesting information, not only through the actual contents of the path but also through the use of meta-information that provides context, pointers to interesting items, narrative point-of-view, and explanation about interface conventions.

These characteristics of a guided path mean that not only experienced users may navigate effectively through a large, complex, and unknown hypermedia system, but inexperienced or novice users may also tour the system and, in so doing, may gain experience that may enable them to later explore it in a more flexible manner. It also enables users to save a particular path through a complex system so that they may re-find their way to particular information at a later date.

Pilot Study

In order to better evaluate the possible use and effectiveness of a guided path facility, a pilot study was carried out. Some of the questions that the pilot study aimed at investigating were:

- Would a path mechanism assist navigation in a large hypertext system, especially for inexperienced users?
- Would naive users navigate better with a path tool than with using HyperCard commands available to them, perhaps due to a reduced learning curve and reduced cognitive load?
- Would expert users explore more — that is, would they go off the path more?
- Would meta-information be helpful through providing contextual assistance, perspective and guidance?

Another aim of the pilot study was to discover what other questions about path mechanisms and general hypermedia system navigation would be worth investigating further.

The pilot study was carried out using HyperCard as the test system. HyperCard is very widely used and a path facility may be valuable in enabling more users to access the increasing amount of materials that are appearing in this system. A path mechanism was designed and built that provided an easy method of creating a path, storing and manipulating paths, and following a path. This mechanism provided two interfaces — one for path creation and one for following a path. This was intended to distinguish experienced users, who would create their own paths and paths for other people, from inexperienced users who would be path followers and would therefore require a simpler interface i.e. they would need less controls because they require less functionality.

Thesis Description

Chapter One offers a general discussion of hypermedia, comprising of a general definition of Hypermedia, followed by a discussion of the origins of hypertext concepts and systems. Next is a short presentation of a number of common hypermedia systems and some of their characteristics.

In Chapter two, navigation and orientation in hypermedia systems are discussed. Firstly, metaphors in the interface are discussed and then the application of the geographical metaphor to navigating hypertext systems (or information '*space*') is explained through various geographical metaphors used in a number of systems. Then the navigational capabilities of various hypermedia systems are presented at three levels — large scale, medium scale, and small scale — as the tools and techniques that are used in finding information differ at these levels.

Chapter Three breaks down the tools that are used for navigating hypermedia systems into specific discussions of their use and utility, with detailed examples from specific systems being presented. The main tools and techniques discussed are Paths and Tours, Maps, Guides and Agents.

In Chapter Four a trail-blazing facility is presented. It begins with a discussion of navigation in HyperCard and follows with a description of the navigational problems that the trail-blazing facility alleviates. Following this is a description of some of the specific mechanisms that the facility uses to help reduce navigation and orientation problems in large hypermedia systems — or more specifically, in large groups of HyperCard stacks. This includes the use of a simple interface and the provision of meta-information.

Chapter Five presents the implementation details of the trail-blazing facility. It explains the structures used, screen layouts, and problems and limitations of it.

Chapter Six presents the Pilot Study that was used to evaluate the trail-blazing facility. Conclusions and further questions are discussed.

In Chapter Seven some overall conclusions about paths as aids in navigation of hypermedia systems are presented. Then it provides some ideas for future enhancement of the trail-blazing facility, as well as some ideas for further research into this area of navigation in Hypermedia systems.

Lastly, the appendices provide details of the pilot study: the HyperCard stacks used, demographic and evaluation questionnaires, a description of how to actually use the Guided Path facility, a program listing, screen dumps of each of the cards that were used in the test path in the pilot study, and finally, the references used in this thesis.

Chapter One

Introduction to Hypermedia

Introduction

This chapter provides a general introduction to hypertext and hypermedia systems. The concepts of nodes, links, and access mechanisms will be introduced in the definition of hypertext. Subsequently, a description of the origins of hypertext systems will occur. This will acknowledge the contributions made by some of the pioneers in the field, such as Bush, Englebart, van Dam, and Nelson. Then a short description of some common contemporary hypermedia systems will take place. The systems covered include HyperCard, Intermedia, NoteCards, Xanadu, the WorldWideWeb, Wide Area Information Servers, and Gopher.

Hypertext Definition / Explanation

In this thesis a differentiation will not be made between hypertext systems and hypermedia systems, although there are, of course, differences between the two. Hypermedia might be thought of as a superset of hypertext because it incorporates more media types. Thus it includes the capabilities and problems that hypertext contains, but it also presents more capabilities and problems due to the inclusion of more media types. Many problems are common to both systems, especially in navigating the information base, so in this thesis the general term hypermedia will be used as this encompasses hypertext as well.

Hypermedia can be defined in many ways. Some previous attempts at definitions include:

> *"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 conventional page."*
> [T.H. Nelson, cited in Conklin, 1987, p.17];

> *"Hypertext, at its most basic level, is a DBMS that lets you connect screens of information using associative links. At its most sophisticated level, hypertext is a software environment for collaborative work, communication, and knowledge acquisition. Hypertext products mimic the brain's ability to store and retrieve information by referential links for quick and intuitive access."*
> [Fiderio, 1988];

> *"A defining attribute of hypertexts is that they embed textual information in related but non-sequential segments and they provide mechanisms for readers to explore this information flexibly, often on the basis of their personal preferences and needs."*
> [Reinking, 1992, p. 19].

The determining characteristics of hypermedia systems are the existence of chunks of information, called nodes; connections between nodes, called links; and mechanisms for accessing the nodes and for following the links from node to node.

Nodes

Nodes are the information base of a hypermedia system. They are the individual documents that are stored in the system and that are displayed in some manner when accessed. The distinction between hypertext and hypermedia occurs in the content of the nodes. In a hypertext system the nodes will be text documents. In a hypermedia system, the nodes may contain text, graphics, sound, animation, video, some other medium, or a combination of these.

Links

Links are explicit connections between nodes or parts of nodes. A link has meaning. Landow (1987, p.332) says that "*Hypertext links condition the user to expect purposeful, important relationships between linked materials*". This means that links are seen as adding meaning to the documents — they provide the structure for the system.

Access mechanisms

A hypermedia system must provide some means of accessing the documents and the links contained in those documents. These access tools must at least allow a user to follow a link and to backtrack — that is to get back to where they came from. Other access tools should be provided to help the reader avoid the disorientation that can occur in complex systems. Numerous access mechanisms should be available so that different types of users can use a tool that suits their goals, experience, and expectations.

Hypertext Origins

The first hypertexts were paper-based. An early example of paper-based hypertext was the Talmud [Oren, 1987]. This was a document that was annotated by successive generations to form a large body of writing with many inter-connections. The principle current paper-based hypertext system is our library system where most documents contain references to other documents. So a library is really a large interconnected body of writing, on a very large scale, however. It is this idea of inter-connections between documents which forms the basis of hypertext but also provides some of its problems.

Hypertext as originally conceived was to operate in a similar manner to the human brain — by association. It was thought that the idea of accessing ideas or files strictly through indexes was far too limiting and that more flexibility was needed to cope with the ever-increasing amount of knowledge in the world. Thus the idea of instantaneous cross-referencing via machine was envisaged by Vannevar Bush in 1945. He called the machine the 'Memex'. He thought that if libraries of information were stored on machine then all that information could be accessed quickly and flexibly. Personal trails could be created through the information and our own thoughts could be annotated over the original material. He proposed that the future would bring "*a new profession of trailblazers, those who find delight in the task of establishing useful trails through the enormous mass of the common record*" [Bush, 1945].

Bush introduced the idea of nodes and links although he did not specify nodes as such. He had the notion that an author (or trailblazer) could find materials that were related through the use of indexes and bring those materials into view on the memex. Then they could create links between the related documents or ideas. These documents and their inter-connections would leave a trail for another user to follow. The other users would travel from document to document via the links, perhaps going off on side-tracks, following the authors train of thought. Thus they might learn from the author's trail, or they could add their own comments for other users. Thus a new mechanism for accessing information was proposed — one that was not directly linear — and it was machine-based which would result in benefits of speed, storage and flexibility. It seemed that it might more closely resemble the way the human mind operates — by association.

Nelson, van Dam, and Englebart pioneered the development of hypertext systems and are responsible for much of the developmental work illustrated in today's systems [Nielsen, 1990]. Their systems — Augment/NLS, Xanadu, the Hypertext Editing System, and FRESS — illustrated many of the concepts of current hypertext systems and have spawned many current research areas, of which hypertext, groupware, electronic publishing and multimedia are but a few.

Englebart's NLS (oN Line System) featured a primarily hierarchical structure, but it also allowed any number of non-hierarchical reference links within and between files. The system had three main aspects which are still often used: a database containing all the text, view filters that would filter the text in the database, and views which structured the display of the selected information.

The system provided view filters for the file structure which would enable a user to control the display of items, as well as to write customised filters in a high-level language that displayed only statements containing the specified content. It has evolved over the years and is still marketed today as a commercial network system (called Augment) that supports a wide range of activities, both individual and group communication, document production, information management, and software engineering.

van Dam and Nelson designed the Hypertext Editing System that was used by the Houston Manned Spacecraft Center to produce Apollo documentation. This was followed by the File Retrieval and Editing System (FRESS) that van Dam and several students developed at Brown University.

FRESS featured a dynamic hierarchy, bi-directional reference links and keyworded links and nodes. No limits were placed on the size of nodes, and multiple windows and vector graphics were supported. As the available technology advanced, the next system, the Electronic Document system, made heavy use of colour raster graphics and extra graphical navigational aids. All this development work at Brown is continuing and most recently has resulted in the Intermedia system which has been an exploration into how hypermedia functionality should be handled at the system level with linking available for all applications [Haan et al., 1992]. It will be described later in this thesis.

Lastly, Nelson is credited with coining the term '*hypertext*'. His idea has been to create an environment where all the world's literature would be online. He called this proposed system Xanadu. The system makes heavy use of links, so that only the original documents and changes to them are stored. Any referencing that occurs is through links that point to the appropriate part of the original document. In this way storage space can be reduced and issues such as copyright and royalties can, perhaps, be more readily dealt with. The development of Xanadu is continuing today but it has not yet been widely used so little information about its capabilities exists.

Some Hypertext Systems

In this section some of the common current Hypermedia systems and a summary of their main features are presented. The systems that are discussed are HyperCard, Intermedia, NoteCards, Xanadu, the WorldWideWeb, Wide Area Information Server (WAIS), and finally Gopher.

HyperCard

HyperCard is a product, originally from Apple Computer, Inc., that has been bundled with every Macintosh system sold since 1987. Because of its widespread distribution, price, and ease of use it has become very popular. Although not originally designed as a hypermedia product, its ease of use, programmability, and rapid linking capability combined to produce a hypertext-like system that could be used to rapidly prototype new design ideas.

HyperCard uses the metaphor of a stack of cards. Groups of cards are formed into stacks. Multiple stacks of cards may be opened at one time and each appears in its own window. Each card may contain a number of fields that can hold textual information. A card can also contain graphics. Thus each card may be thought of as a hypermedia node. Only one card from a stack appears in each window at a time. However, pop-up fields can simulate the appearance of multiple nodes being open at one time. Another component on a card is a button that usually contains some code or Hypertalk script. Hypertalk is HyperCard's programming language. Hypertalk scripts can perform almost any task because the language is extensible through the use of external commands and functions called XCMDs and XFCNs. These can be written in languages such as Pascal and C and provide a way to access lower level functions within the Macintosh Toolbox.

The provision of an extensible language provides much power to HyperCard, and enables complex links to be set up between different objects such as cards, buttons, fields and text within fields to other objects. This means that very complex documents or groups of documents can be formed. The language and extensions provide access to a range of media including sounds, 2- and 3-dimensional graphics, and video. So it is a potentially very complex system. Usually a HyperCard stack will be on one topic and the amount of information contained within it will be relatively small. However, more complex stacks are being developed, as well as groups of stacks, and these can present navigational problems for some users.

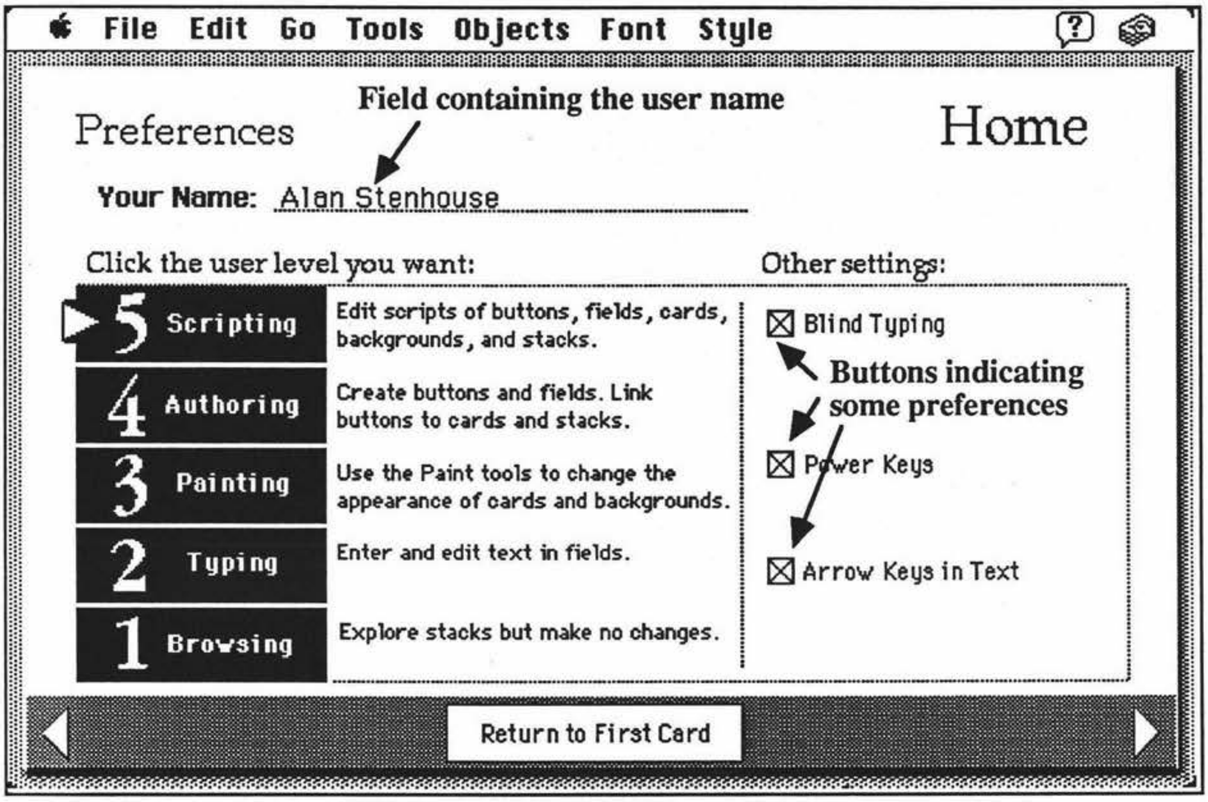


Figure 1.1 *An example HyperCard card showing fields and buttons (from the HyperCard Home stack).*

HyperCard provides a number of intrinsic navigational facilities, although other facilities may be provided within a stack through the use of the Hypertalk language. The intrinsic navigational facilities are found in the Go menu and are listed in Chapter Four. Two that are significant are the Recent map and the Go Back menu command. The Recent map provides an iconic view of the 42 most recently visited cards. Selecting one of the icons results in a return to that card. The view of the cards is not in strict temporal order, however. If a card is visited more than once, only the occurrence of the first visit to the card is added to the map. The facility that provides a strict temporal history is the Go Back command. This uses a temporal history that can contain up to the last 99 nodes visited, in the order of visitation.

HyperCard is a very useful tool and its power, as well as many of its problems, stems from its great flexibility. The application of design guidelines as well as the creation of some extra facilities can alleviate many of the problems associated with navigating groups of stacks. However, for it to become a real hypermedia system, operating system-level support is required to provide basic hypermedia services.

InterMedia

Intermedia is a hypermedia system developed by the Institute for Research in Information and Scholarship (IRIS) at Brown University. The intention of IRIS was to "*create a model for how hypermedia functionality should be handled at the system level, where linking would be handled in the same way for all applications*" [Haan et al., 1992]. Intermedia provides a graphical file system browser similar to the Macintosh Finder; a set of direct manipulation editors for text, graphics, timelines, animations, and videodisc data; a set of linguistic tools; and the ability to create and traverse links in any document in the system. The system uses privileges to control access to documents so there is no distinction between authors and readers. If a user has the appropriate privileges then they can edit, explore and annotate as they wish.

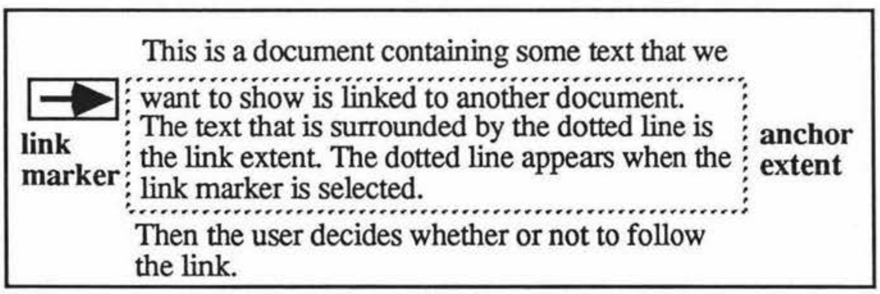


Figure 1.2 *Link marker and anchor extent (similar to those used in Intermedia).*

All the applications support anchors, anchor extents, and links. The anchor is the link origin, the anchor extent is the block within the document that is linked, and the link connects the block to other blocks. See Figure 1.2 for an example. All links are bi-directional so a user can traverse a link and then easily return to where they came from. All applications support a consistent visual appearance of the link marker which is a small icon positioned to the left of the anchored block. This is an important stylistic convention that immediately conveys the knowledge that it is a link and can be clicked upon. This is very important for small and medium scale navigation as the consistent approach removes some cognitive overheads for the user. That is, you don't have to spend your time searching around for links — you know immediately what is linked and what isn't by the presence of the small link icon and the link icon is the same throughout all Intermedia documents.

The idea of an active anchor was added to provide extra functionality to temporal data such as video and animation [Palaniappan et al., 1990]. An active anchor includes an action flag so that, on following a link, the time-based data (such as video) will run if the flag is set. If the flag is not set then the destination block will be highlighted and can then be viewed manually.

Sets of links are organised into webs that may be browsed using the Web View. The web view facility provides an organised visual map of the network of links and nodes to give the user some sense of context and location within the web. The web and the web view will be described further in Chapter Three — Navigational Aids in Current Systems.

A distinctive feature of Intermedia is that information about the anchors and links is maintained in a database management system separate from the document data [Haan et al., 1992]. Because editing operations may be carried out by anyone with the appropriate privileges it is very important that the system be able to maintain the integrity of the links and their respective anchors. This is necessary to prevent the occurrence of "dangling links" — links that refer to data that has been deleted. The separate DBMS ensures data consistency. All editing operations performed by the applications or by the file system browser pass through the IRIS hypermedia services which coordinate updating. For more information on the specific mechanisms employed see Haan et al. (1992).

Intermedia has been used reasonably widely. Professors in English, Biology, Anthropology, English as a Second Language, and Geology have prepared webs for their students. Other assignments have been for the students to create their own webs through the material and provide commentary and annotation. In addition, some students have been encouraged to link comments on each others' work [Utting and Yankelovich, 1989]. Benefits from using the system seem to be rapid and quantifiable. Students using Context32, a web on English Literature, gave far more detailed and intellectually sophisticated answers in assignments and tests than previous students who had not used it. In addition, inexperienced students made better use of navigational aids, such as introductions, footnotes and glossaries, in books after realising the inter-connectedness of literature and realising the importance of these connections [Landow, 1989].

In summary, the Intermedia system provides a number of tools to support navigation but that these tools primarily deal with medium scale navigation. Large scale navigation and small scale navigation are not particularly supported by the system and are largely left to the user to organise. These might be areas that could be looked at for future implementations. [Note that IRIS is soon to be discontinued due to funding withdrawal so further development of Intermedia seems unlikely unfortunately (Livingston, personal communication).]

NoteCards

NoteCards is a hypermedia system developed at Xerox PARC designed to help people work with ideas. It uses the concept of an electronic notecard — an idea represented on a small card that can be combined with other notecards in various ways to form an orderly and logical argument or presentation. The system provides a variety of tools that enable users to collect, interrelate and communicate ideas and then to display, manipulate and navigate the network of ideas they have created [Halasz, 1988].

A Notecard may contain an arbitrary amount of information such as text, a drawing, or a bitmap image. Each card also has a title. Every card can be edited. There are also different types of notecards depending in part on what they contain, but new types of cards can also be created.

The links that are used to connect these cards are typed — they have additional meaning other than just in connecting two ideas linearly — and directional — the link is from one card to another card and not necessarily both directions. An example of a typed link is a supports link, which is used to connect underlying reasons to a conclusion. This shows that because of the link typing the link is uni-directional. Although, of course, the opposite direction could be said to mean *is supported by*. These links are anchored at a particular location in the source card but point to the whole destination card. The link anchor is represented by a small link icon that can be clicked on to traverse the link.

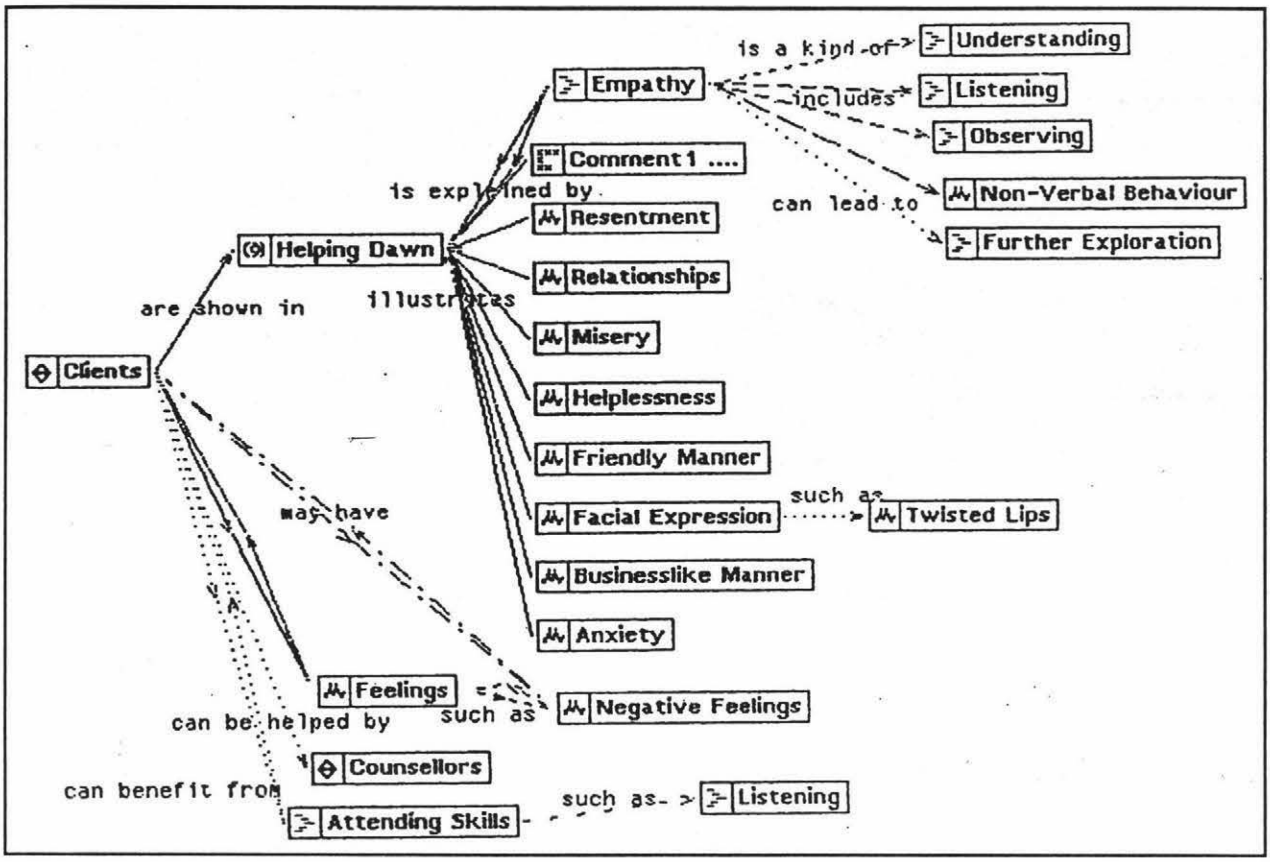


Figure 1.3 *A Notecards Browser card showing link typing (from [Duncan, 1989]).*

The two most basic tools to manipulate these notecards and links are specialised types of cards called browsers and fileboxes. A browser is a card that contains a graphical display of the structure of a network of notecards. Cards from the network are represented by their title in a box. Links are represented by lines between the boxed titles. Different line styles represent different link types. The browser enables direct access to any of the cards included in the display through clicking on the card's boxed title, and thus navigate the network. The graphical display is constructed automatically by the system and, once created, can be used to modify the underlying structure of the network of cards.

A filebox is a specialised card that can be used to organise large collections of notecards, mainly designed to help in the management of large networks and to aid storage and retrieval. A filebox is a card in which other cards, including other fileboxes, can be filed. All notecards must belong to one or more fileboxes.

Fileboxes were designed to help users manage large networks of interlinked notecards by encouraging them to use an additional hierarchical category structure for storage and retrieval purposes [Halasz, 1987].

To access information, a user primarily navigates by following links from card to card. These links are distinguished from normal text and graphics by surrounding the linked text with a box and emboldening the linked text or by providing a small link icon. This convention is used throughout the NoteCards system to designate a link, and is therefore immediately recognisable and easily used. An alternative method of navigation is by using a browser card to traverse the network directly. All nodes on the browser are links that lead directly to the specified notecard. Here again, the nodes on the browser card are distinguished as links by the convention described previously. NoteCards also provides a limited search facility which will locate all the cards that match some user-specified conditions.

The NoteCards system is quite customisable, but only with some difficulty for many users. It is written in the Xerox Lisp environment and is fully integrated into it. This gives much potential power for extending NoteCards functionality, including being able to integrate it into another Lisp-based system such as an expert system. The system also has many customisation options with which the user can set to tune its exact behaviour (e.g. how links are displayed or the default size of notecards). [Halasz, 1988].

Other types of notecards, such as TableTops and Tours, have been built. These will be discussed in Chapter Three — Navigational Aids in Current Systems.

Xanadu

Xanadu's implementation of a large hypertext system has initially been concerned with the underlying structure and the complexity of the addressing schemes used. The main problem is that the system must keep track of an ever-growing number of items, which means keeping track of many numbers in the form of addresses. The *'docuverse'* might become very large in an unpredictable fashion, so they were concerned with forming an addressing scheme that could deal with this, but also with the idea that it would be concise and clear when used 'in the small' — with individual documents.

Inspiration was taken from the Dewey Decimal System concept of forking numbers that lead to the idea of Humbers or humungous numbers [Nelson, 1988]. They are numbers that can be continually separated to form more numbers. Such an addressing scheme implies a tree or hierarchical structure but this is only on the address space of the system, not on the materials contained within the system. These humbers contain four major fields: Server, User, Document and Contents.

The Server field is the address of the node where the document is stored. The User field is the address of the owner or the document. The Document field is the address of the logical entity in which the material is stored. The Contents field is the address of the actual document contents which can be the actual bytes of material or can be a link to another document depending on the first byte of the address. Protocols have been defined to handle the underlying connections so that the work of an application designer becomes largely a matter of the user interface.

At the moment not enough information is available about proposed access or navigational mechanisms, but with the underlying structure being well defined, it should be less of a problem to provide navigational tools for the system. Some of these tools should be provided as part of the system, in order to present a clear and consistent interface to developers and users of applications that make use of the Xanadu architecture.

World Wide Web Project

The WWW project provides a browser program to access information anywhere in the world [Berners-Lee, 1991]. It uses a simple protocol ("HTTP") but is also available using other existing protocols such as File Transfer Protocol (FTP) and Net News Transfer Protocol (NNTP). It utilises the Standard Generalised Markup Language (SGML) to create webs. These webs can then be accessed using the browser program, either by following a link directly by selecting it, or by searching indexes for keywords. The links are addresses of specific documents, but the documents may be located anywhere in the world — the addressing scheme enables this. The addressing scheme includes information such as the machine address in a certain protocol format, the directory path where the document is located and the document name itself.

Although the scale of the project is large, the structure of the project is conceptually simple. There are, therefore, the very simple navigation tools of a basic index search and linking mechanism provided with the system. This is only a beginning for the system. If this could somehow be integrated with other medium scale systems and the medium-scale tools integrated with these conventions, then a very powerful system would result. At present, it has great potential but is of limited use to anyone except those that are technically oriented already.

Wide Area Information Servers (WAIS)

WAIS is an attempt at a large-scale hypertext system by allowing links to be deduced at run-time and across many databases stored in many places [Kahle, 1989]. Through a simple string pattern-matching query mechanism, servers provide pointers to relevant documents ranked in importance. These document pointers can be thought of as a form of hypertext link — they can be put in another document and retrieved at a later time. By combining many people's groupings, one can navigate through large numbers of documents in potentially interesting ways in a hypertext style.

So this is different style of hypertext linking — it is content- and user-driven rather than having a specific author/reader dichotomy. This would seem to be one of the characteristic features of large scale hypertext systems when compared to the other levels. The large scale systems are aimed at the users forming their own links for their own use. The distinction between author and reader becomes blurred as no special skills are required to create links to other documents. This, of course, does not mean that the quality of the links created are any good, and this is where the value can be added by the information '*trailblazers*', as Bush (1945) described them.

Gopher

Gopher is another wide area information system based on client/server architecture that began as an attempt at a University campus information system. It consists of an interface to the many pre-existing facilities available on local and global networks. Through the interface, links can be set up to sites or individual files around the world. It hides the complexity of link information behind a graphical interface similar to the Macintosh Hierarchical File System (HFS).

Nodes are currently coarsely grained — they are a geographical site or an individual file. Inter- and intra-document links are not supported at the moment. When the granularity of the system enables these links then the system will be on its way to true hypertext, although much more needs to be done to the system in order to support finer granularity. Additional access and management tools are also needed to provide navigational assistance at many levels. At the moment, navigational management is left up to individual users. The system does not automatically assist management of related materials.

In the near future, the integration of facilities such as Gopher and WAIS will provide a powerful global information system. In their current forms, however, they are not highly usable by unskilled users. They also do not provide support for inter- and intra-document linking which is one of the bases of hypertext. However, tools might be provided that can be combined with these systems to provide the capability for automated or manual linking between nodes. System support of the links is needed, however, in order that the integrity of the links is maintained in the light of deletion and modification of nodes.

Navigation and Orientation in Hypermedia

Introduction

This chapter describes current methods of finding information within hypermedia systems and describes some of the problems associated with this. Finding information is not only a matter of accessing the correct location, often it is necessary to understand the current location before accessing another location. This is a problem of orientation, and the problem of navigation follows on from this. The process of finding information can be viewed as a cycle, with some navigational activity occurring, then some orientation takes place, and depending on the current location, some more navigation may occur, and the cycle continues.

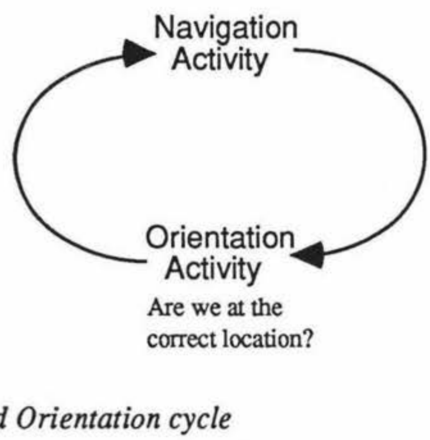


Figure 2.1 *Navigation and Orientation cycle*

So the idea of navigating a system to find information within it is closely linked to the idea of orienting oneself within the system. It is a continuous process of action, feedback, and evaluation.

Hypermedia systems can be very complex and many different facilities exist to enable users to navigate and to orientate within the systems at different levels. In this chapter, the use of the geographical metaphor within computer systems and its application as an aid for information search is discussed. Then a summary of the facilities that are used at different levels of information navigation is presented. Finally, there is a discussion of some of the problems of navigation and orientation within hypermedia systems. These problems include indicating where links exist and what their destinations are, indicating scope at various levels, providing effective backtracking mechanisms, and differentiating meta-information from node contents.

The Geographical Metaphor

People use metaphors in everyday life to make it easier to understand little-known concepts. They are an integral part of everyday thought. People try to understand unknown concepts through comparing them to something known that might exhibit similar characteristics or behaviour. "*Metaphors function as natural models, allowing us to take our knowledge of familiar, concrete objects and experiences and use it to give structure to more abstract concepts.*" [Erickson, 1990]

Various metaphors have been used on the computer. A successful example is the desktop metaphor used on the Macintosh computer. It attempts to hide the complexity of the operating system by presenting a consistent, familiar image to us. It represents our electronic documents and applications (or tools) as icons that appear similar to small pieces of paper. These documents and tools are available on a desktop. Tools and documents can be stored in folders on the desktop, and they may be moved around by directly manipulating them. A document can be thrown away by placing it in the trash can. Of course, all this is a metaphor — the electronic documents do not actually move, pointers to them are moved, but this would be too complex and difficult to understand for most users, so the interface metaphor works to a large extent.

Some exceptions continue to cause problems, such as the ejection of a disk by dragging the disk icon to the trash-can. This is where continuing the metaphor has problems — what can a disk be associated with on a real desktop? Perhaps the disk should be ejected by placing it in a filing cabinet or disk box? This would be continuing the desktop metaphor if the disk was thought of as a folder to be filed away, whereas putting it in the trash would seem to be deleting what was contained within it. The success of a metaphor is attributable largely to the correspondence in appearance, use and behaviour of the interface 'objects' — documents and folders — and their real-world counterparts. Often users' problems are in the differences in behaviour, or where the 'mapping' between objects or situations breaks down. Successful metaphors should emphasise certain features and suppress others [Mountford, 1990].

Information Space and the Geographical Metaphor

The quantity of information that is available is often likened to the 'world of information', 'information space', or the 'information landscape' [Florin, 1990]. This world of information has often presented problems for people who want to find information within it. The previous physical nature of much information — that printed in books — and its subsequent transferral to electronic media has resulted in the actual geographical problems associated with finding information in the physical world being compared, through metaphor, to the problems of finding information using the computer in the electronic 'world'. The nature of finding one's way in the real world is also likened to that of finding information, because a place in the world *is* information — it is actual physical substance and not an abstraction through words or symbols.

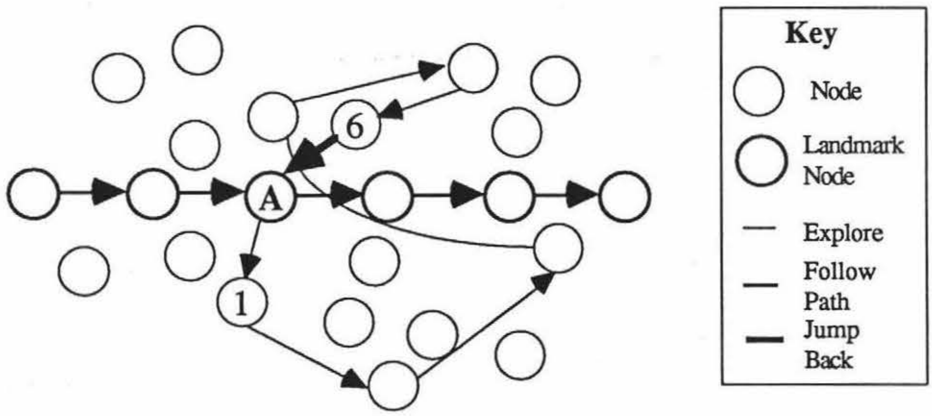


Figure 2.2 *Following landmark nodes along a path until node A. Then exploring off the path from node **1** to node **6**. At node **6** a jump back to the landmark node **A** is made, so the path can continue to be followed.*

The appearance of hypermedia systems has brought increased usage of the geographical metaphor for accessing information. The concepts of nodes and links have resulted in metaphors such as *following* a link, and when a number of links have been followed, a *path* has been followed. *Landmark* nodes are often created that can be rapidly visually distinguished from the surrounding nodes or space. This is obviously using a concept that is familiar to almost everyone — the idea of being or going somewhere, and this in turn uses the idea of landscape or geography.

> *"The terrain on which the information landscape is built is the raw database, rich with various materials ... the information structure is what gives the landscape its distinctive features."*
> [Florin, 1990, p. 31]

Parunak (1989) describes five different topologies that might be used in the hypermedia world and describes the navigational strategies that would apply to each topology. From simple to complex, these topologies are: linear, hierarchy, hypercube, directed acyclic graph, and arbitrary. Different users apply different strategies when navigating these topologies. The main navigational strategies used in the real world, and which may be applied to other complex systems such as hypermedia systems, are:

— Identifier Strategy — associates a unique identifier with each entity of interest thus permitting the searcher to recognise the target.

— Path Strategy — provides a procedural description of how to get to the target.

— Direction Strategy — uses a global framework as well as the ideas of texture and comparability. 'Texture' is the existence of a reference point relative to which directions can be established. 'Comparability' is the existence of a relation for any two points in the space.

— Distance Strategy — bounds search to a circle (based in time or distance) around the traveller's current location. Often used in conjunction with the direction strategy, but becomes degenerate when any point is accessible directly from any other point.

— Address Strategy — refines the direction strategy by establishing an orthogonal set of coordinates, such as longitude and latitude, or the grid formed by streets in a town.

Of course, these strategies are often employed together. The number of these strategies that can be used reduces as the complexity of the topology increases. So when there are complex topologies as in many hypermedia systems, ways must be found to reduce the complexity so that more strategies may be utilised in order to effectively navigate to items of interest.

One way to do this is to impose some other structure on top of the system. Another way, which is complementary to additional structure and continues the geographical metaphor, is to use tools that are familiar to us from real-world navigation, such as a map or a beaten-path mechanism. A beaten-path mechanism might just be a history function that remembers the nodes that have been visited and allows the user to backtrack easily. A map can be a two-dimensional representation of the information landscape and can be used at a number of levels as in physical geography.

Other geographically based metaphors have been used in some circumstances. Hammond and Allinson (1987) used a tourist and travel holiday metaphor as a design aid in helping users to navigate complex systems. They have extended the idea to learners getting on a 'tour bus' to follow a guided tour through the system [Hammond and Allinson, 1988]. The tourist and guide metaphors have also been used by Fairchild et al. (1989). The systems that these have been incorporated into have all been relatively small. A large scale system that utilises the geographical metaphor is the Hyper-G system which extends these ideas into the 'world of travel' metaphor [Davies et al., 1991]. This enables them to bring many different metaphors — such as maps, guided tours, paths, agents or guides for varying purposes (e.g. travel agent, tour guide) — under one unifying theme. This can help users to relate what they know about navigating the real world to navigating the world of information. When they need help finding something they might go to a travel agent who can suggest where to go to find it, or the agent might suggest a guided tour of an area if the user is a novice. There might be information guides, similar to travel guides, which give advice about what to see, where to find it, and what value it has — this is like getting an expert's opinion on an area and it's probably more valuable than asking your next-door neighbour.

The path facility described in this thesis continues the geographical metaphor. A guided path is used as a mechanism with which anyone may tour through information in a structured way (i.e. following the path). Each node can be considered a landmark, and each link can be considered the path between landmarks. At any landmark side trips available from it can be taken, and then the landmark can be easily returned to when the side trip is finished so the tour can continue. Thus the structural and conceptual simplicity of a linear path can be combined with the possibility of structurally complex side trips at any point.

Navigating Hypermedia — Overview

How is information accessed in a hypermedia system? There are many methods and tools that are employed which can be classified into three main groups: large scale, medium scale and small scale. The large scale tools are used with large-scale hypertexts — those which encompass multiple sites or ones that span the globe, for instance. Medium scale tools are used with medium scale hypertexts — those based at a single site, for example. Small scale tools are those which are used to navigate a single hypertext document.

Large Scale Hypertexts

In the past, most effort has been aimed at developing tools for navigating small and medium scale hypertexts, mainly because those are the main kinds that have been developed so far. There are very few, if any, large scale hypertexts as yet, though the problems inherent in large scale hypertexts are being considered. The most ambitious large scale project is Ted Nelson's Xanadu project which aims to be "*servicing hundreds of millions of simultaneous users with hypertext, graphics, audio, movies, and hypermedia*" [Nelson, 1988].

Another large scale project is the WorldWideWeb (WWW) project being developed in Switzerland. Its aim is to increase the accessibility of academic information, "*... to allow information sharing within internationally dispersed teams, and the dissemination of information by support groups.*" [Berners-Lee, 1991] It uses tools such as indexes, search facilities, and browsers. All documents look the same to the reader and may be accessed in two ways — through an embedded link, or via a search mechanism. It is stated that "*These are the only operations necessary to access the entire world of data.*" [Berners-Lee, 1991]. That may be true, but in order that the system be readily usable, other orientation and navigational support tools must be built into it. Merely having access to data is quite different from being able to access it easily. It also ignores the need for visual cues that are useful in the various levels of navigation.

A very recent project is the Wide Area Information Servers (WAIS) project [Kahle, 1991]. This uses both static and dynamic linking to access information from all over the world using a standard protocol. Navigation is achieved through a search mechanism using a natural-language type query. The words in the query are matched to documents, with articles that have the most words matching being ranked highest in importance. There are a variety of other navigation mechanisms being a variation of this main theme.

Other navigational tools could be provided so that these systems are more easily usable. For example, a map or globe facility would be helpful in determining a number of things about the information, such as cost, and time to access it. Details such as these are important for navigation in the large because if we are going somewhere to get something, it should be known how long it will take and how much it's going to cost.

Medium Scale Hypertexts

Medium scale hypertexts offer a wide range of navigation and orientation tools, and these tools vary from system to system in their implementation. Almost all, however, offer the two main types of tools — a graphical browser, and a search facility. These vary widely in their implementation, however, and there are also a number of other tools used for navigation in medium scale hypertexts. These tools will be illustrated from two of the most well known current hypertext systems — Intermedia and NoteCards. Subsequently a few other tools or tool refinements that could be useful for these systems will be suggested.

Graphical Browsers

Most current hypertext systems offer some type of graphical browser with which to navigate the system. It is somewhat analogous to a paper map although on the computer more flexibility exists than on paper so most browsers are automatically generated to give a sense of context. In some manner they attempt to show current location, past locations, and possible future locations. In the past, some browsers attempted to show all nodes and links in the hypertext, but this was soon found to be unrealistic because as the number of nodes increases, the number of links increases exponentially [Baird, 1988]. This quickly makes for spaghetti-like maps, so some form of selection and filtering must take place in order that a browser be usable by an individual.

Another problem with browsers is that they seem to be needed on a number of levels. First there is the local browser which shows your immediate context within some document. Then there is an overview browser which might attempt to show medium-scale context — that is, it might show the nodes at the current site and some linkages between them. Then perhaps there will be a need for a global map, which shows how systems around the globe are linked.

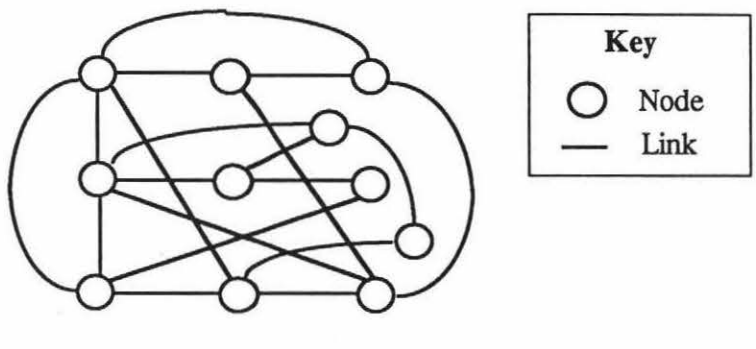


Figure 2.3 *A spaghetti-like map begins to take shape if a graphical browser does not filter the nodes and links, especially as the number of nodes and links increases.*

The first two browsers could perhaps be implemented in one using some sort of 'fish-eye' browser [Furnas, 1986], which shows a high amount of detail for nodes in the immediate vicinity, but the detail becomes increasingly less as the distance from the point of interest increases. So a sense of our immediate context can be gained, but also some idea of context in a larger sense. Local detail can be seen in a global context.

The global browser may be more difficult to implement but might be helpful in a number of ways. To begin with, it would serve to give an indication where repositories of certain types of information were located. It could also serve such purposes as 'information demographics' much as a geographical map displays geographical information. Perhaps the map could change to display relative densities of various types of information. Hence if a particular area was interested in, a filter could be applied to display where the main centres for this information were using the global map. Then perhaps a particular area could be zoomed in on and the overview browser or map for that area would appear. This might enable us to quickly locate areas of relevant information. It is a method of filtering the information displayed so that it is relevant to us. Thus as the filter is defined more precisely, the displayed map becomes even more meaningful. The 'zooming in' to display different levels of a map or browser illustrates the multiple levels needed in order to get immediate, local and global context. There can also be different levels within these arbitrary divisions.

The information shown in these graphical browsers can vary. Some are text-based with each node being represented by a short description of its contents. Some are graphically based with a node being represented by an icon. Generally these icons represent the type of node, for many systems allow typing of nodes, e.g. a node might be text, graphic, sound, video, animation, or some other type, and each will have its own iconic form on the map. Often there will be a combination of the two with an icon representing the type of node and a short textual description of it. Of course, in a hypermedia system, each node should probably have a short representation of its contents. For example, a graphic might have a thumbnail of the graphic, or a node containing video might display a short small video clip. This would conform to the user-interface principle of progressive disclosure, although in a system with an unlimited 'go-back' facility and quick response time this might seem less necessary.

Intermedia offers an interesting and valuable tool called the web [Utting, 1989]. This is a device that helps us get context by filtering out unnecessary information. A web can be opened and within it only those links and nodes that belong to that web will be available. Therefore it simplifies the graphical browser, especially where multiple links emanate from nodes, as it can only show the nodes and links that are applicable to this web.

It might also be possible for us to create our own webs so that when the web is returned to at a later date, a similar sense of context can be gained through following the same path that was taken previously. This would seem to be quite desirable, for although the seemingly unstructured nature of hypertext is deemed as one of its major strengths, when an argument is developed a line of thought is followed. So a web that replays a linear path through the structure can help as a reminder of what was previously gained from it. Or it could be used to apply linearity for someone else to read. Then it is a form of guidance which may be used to help a novice through the difficulties involved in navigating complex structures.

Search Mechanisms

The other main medium-scale navigational aid is a search mechanism, which enables users to break through structural boundaries to locate information. A search is often used when a particular item, which satisfies some criteria, is being looked for and its location is unknown. So a rapid scan of the system may be used to check for the existence of the required items. Often an index will be used within the search mechanism. A search results in a number of node hits which can then be evaluated as to relevance, or the nodes can be immediately accessed. A hit can be classified as an occurrence of the item that satisfies the search criteria.

In the Dynamic Medical Handbook Project [Frisse, 1988] a node's relative value is calculated from two components. The intrinsic component is calculated from the number of hits within that node, while the extrinsic component is a value computed from the weight of the node's immediate descendants. The sum of the two components gives a node's value that can then be displayed, perhaps on the browser, so the nodes which are the most relevant to our query can be immediately identified. In this case the query could serve as a filter and display only those nodes that are hit. When these individual nodes are accessed, any descendent nodes from there may also be accessed, not only nodes that contain hits.

Other search mechanisms use various weighting methods to assess the relevance of a particular document or node to the search query. The WAIS system uses simple text pattern-matching algorithms so that the number of hits of the same text within a document will be reflected in its ranking [Kahle, 1989]. Other systems use sophisticated searching techniques that can also base searches on the structure of the system. For example, when a system uses typed links, as in NoteCards, the user might like to find nodes that both contain some particular text and are connected to other nodes by a particular type of link [Halasz, 1988]. This is a very powerful search mechanism, but one that requires much knowledge about the system by the user so it is more relevant to specialised systems. This does not mean that it shouldn't be provided in all systems of course. (For an extended description of search techniques, see [Salton, 1989] or [Ellis, 1990].)

Landmarks

The Landmark is an important tool used in medium scale hypertexts. It provides navigational information through being a recognisable reference point from which a sense of orientation can be gained. That is, it can be easily returned to and, from it, other previously visited nodes can be returned to. This illustrates a need for having regular landmarks throughout a hypertext, and there are a variety of ways to make a node a landmark.

One way of doing this is to use different coloured or textured backgrounds for an occasional node. Of course, for this to be effective it should not be random. Rather it should be a node whose contents are somewhat different from its surrounding nodes. It could, for example, be a section heading. Nielsen [1990a] uses different textured backgrounds and different graphical designs in his hypertext to distinguish a regular node from other elements of the system.

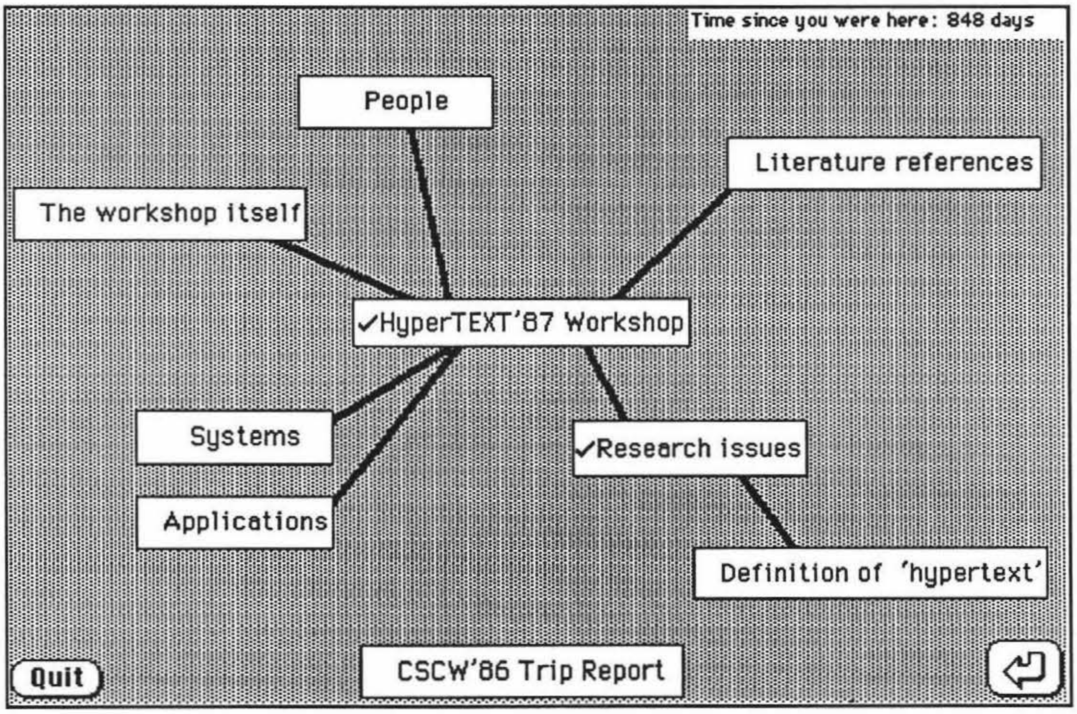


Figure 2.4 *Example of a Landmark Node. This is the main overview for Nielsen's Hypertext '87 Trip Report. It is immediately recognisable and is easily accessible from any part of the report . In addition, this landmark provides an indication of the structure of the report*

Another way of providing a landmark in medium scale hypertext is to use reference nodes. These might be nodes which are in some way relevant to the subsequent nodes, a section heading or table of contents for example. They might be nodes at an upper level of a hierarchical structure. Often part of the reference node might be carried on to subsequent child nodes — subsequent nodes of the same section — to indicate the underlying structure. This might be the carrying over of a section heading, with it being in a consistent location throughout the section. This would assist the user in orientation and thus reduce the cognitive overhead involved so the user can concentrate on the content and not on working out the structure or what the relationships are.

A reference node should be a node that can be returned to very easily and which is immediately recognisable through some distinguishing characteristics. These distinguishing features might be visual cues such as colour, spatial layout, and background pattern, or perhaps it could be a feature such as an audio sequence, or animation, or more likely a combination of these. A combination of cues is needed because as a system gets larger, multiple cues are needed to distinguish one landmark from the next [Nielsen, 1990a].

Table of Contents

A table of contents seems to be another valuable tool for medium-scale hypertexts, because it presents an introduction to the system contents so that the user can quickly determine whether this area is relevant. It can also aid in conceptual understanding and the construction of mental maps of the system structure [Simpson and McKnight, 1989] which can increase navigational efficiency.

Index

An index is also a valuable aid as it can provide direct access to an item that is known to be in the system, when it is not known exactly where in the system it is. This form of index is similar to an index in a book, which contains items that are deemed to be of interest by the author and pointers to their locations in the document. It differs from the index associated with a search mechanism in that a search index will often contain all information in the system in an indexed form, and this will not be in a form that can be directly used. It must be accessed through the search interface which will allow the user to construct a query. The book-like interface will contain the author-specified terms of importance and their locations, and these terms may then be directly accessed within the system.

Small-Scale Hypertexts

Navigating hypertext in the small uses a variety of tools and techniques, most of which are visual cues that aid our understanding of the hypertext structure, node content, and screen layout. Thus they are aids for understanding the current position in relation to the medium-scale (i.e. the current structure), how to find information on the current screen, and how the current screen's information is structurally arranged. Standards also have an important part to play in small-scale hypertext navigation because, once the standards are learnt, they allow the user to concentrate on the information presented rather than dividing the user's attention between presentation and content.

Thus small scale tools are primarily visual cues and conventions, much as standardised conventions such as page numbering and paragraph arrangements exist in printed media. Kahn et al. (1990) identify three graphic design principles that are appropriate for the design of hypermedia documents and relevant to small-scale navigation:

The rules of type — the relationship of type, leading, and line length to legibility. To maintain legibility, adjustments must be made to account for the low resolution of the computer screen.

Consistent formatting — single publications or series of publications should contain consistent formatting rules to support reader orientation.

Clear information graphics — again the low resolution of the computer screen should be taken into consideration.

However, a hypertext offers not only more flexibility than normal documents but also potentially more problems. With the flexibility of presentation come additional problems in understanding what is presented. One particular area of interest is the question of whether to present many windows per screen, or only one window per screen. An extension of this question is whether to present one idea on a scrollable window, or whether the window size should be fixed and if the idea is too large for one window, then it should be split across multiple windows.

If one node is presented as a single idea, how should this be implemented? One of the main aims should be to make it clear how much information is contained in the node so how much more there is to read, look at, or understand is easily known. The implementation of this might vary. One very common way is to present a paragraph on a single screen, with many screens making up one idea or concept. Another way would be to have a scrollable window presenting one concept on one screen.

The main issue involved here is the presence of cues that enable users to ascertain how much information there is, how long it might take to go through it; how a previous position can be returned to; and how to find something of interest within the idea presented.

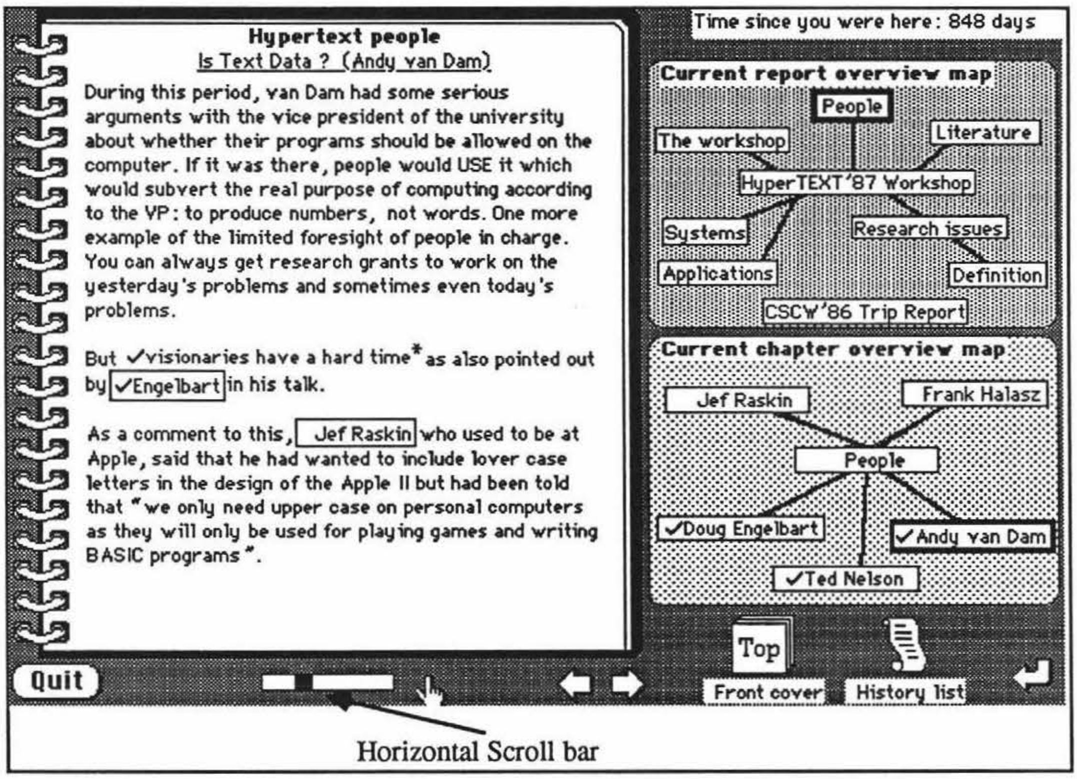


Figure 2.5 *Visual cues indicating structure. Here a horizontal scrolling mechanism (the white bar with the small black box in it) can be seen at the bottom left, indicating that the current screen is about number 3 out of a total of 8 screens for this section. At the right, structural maps can be seen, indicating the current chapter's position in the document, as well as the current section's position in this chapter. Note that navigational and orientational aids occupy about one half of the available space.*

In the many screens per idea approach, page numbering schemes can be used as a cue, with an example being the words 'Page 1 of 6' at some position on the screen. Or a visual cue such as that used by Nielsen in his Hypertext '87 Trip Report [Nielsen, 1987] could be used. He uses a version of a horizontal scroll bar to indicate the relative position through the current idea. A type of horizontal scroll bar can be used to move forwards or backwards to a position, allowing rapid re-positioning through the hypertext. It is also possible to move linearly through this section of the hypertext using the next page and previous page icons.

In the one-screen-per-idea approach a scrolling mechanism might be used with other visual cues to indicate position. One way would be to use modified scroll bars such as used in the Atari computer rather than those used on the Macintosh. These indicate the amount of information being currently presented relative to the whole so that how much information there is, where the current position is within it, and how much more there is to go within this node, is easily ascertained.

One possible problem with the one-screen-per-idea approach is that the micro-cues that are obtained from a single page may be more difficult to understand. For example, if it is desired to quickly browse through a section searching for a previously known location, then cues such as page layout and relative white space (the layout of paragraphs and sentences), that can distinguish one page from the next, have probably been lost. With the many pages per idea approach more cues are available so it becomes easier to navigate. This assumes of course that the cues presented are unambiguous.

Summary

The flexibility offered in accessing information through hypertext systems must be accompanied by various tools that enable all levels of users to access the information efficiently and effectively. There are a number of levels of navigational tools and cues that must be provided for such access to occur, and these can be categorised as large-, medium-, and small-scale. Some concepts are present at all levels, such as landmarks and queries, while others are particularly suited to one level or another, such as a table of contents which is used for medium-scale hypertexts. Many current systems provide medium-scale tools because the systems are designed for that level. The small-scale presentational conventions are usually left up to the individual designer and this is somewhat unfortunate as a multitude of styles leads to incompatibility and difficulty in understanding. The systems that are designed for the large-scale usually neglect the medium- and small-scale navigational aids, and these need to be provided for these systems so that accessibility and ease of use are increased. An understanding of the multiple levels of navigational mechanisms is required to provide effective usability for the differing requirements of many users with differing skills and experience, and to provide efficient access through the different levels or 'scales' of hypertext systems.

Problems in Navigating Hypertext Systems

As with any system, casual users will often have problems with basic tasks. In a hypertext system, the main problem that users have is one of navigation. This manifests itself in questions such as:

Where am I?

What can I do here?

How did I get here?

Where can I go, and how do I get there? [Nievergelt & Weydert, 1980].

Is there anything about 'X' here?

I know there's something about 'X'— how do I find it again?

How much is there here?

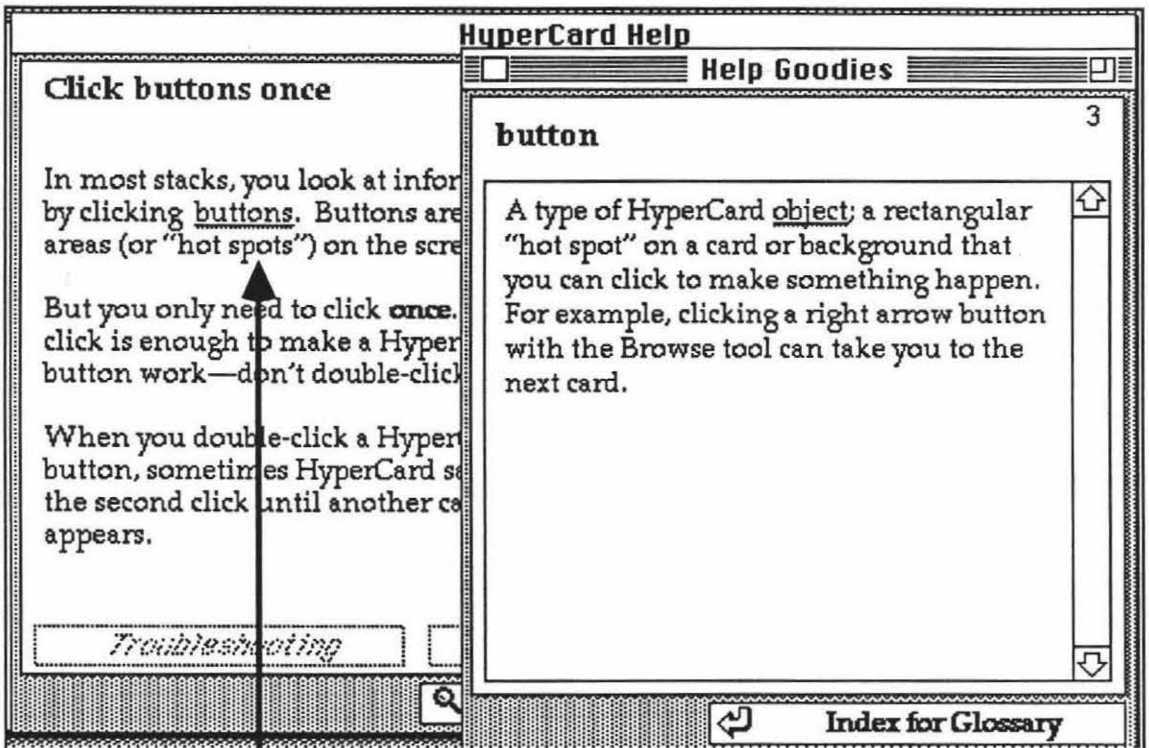
These problems are to do with navigation and orientation, and can result from a variety of deficiencies in the system itself or in the design of the hypertext documents.

It is very difficult to structure a hypertext system so that it can satisfy all users and all tasks. Consequently it is often difficult to provide an appropriate overview of the material — some users may fail to see the structure of the information base, and so may miss out complete sections. The number of potential links and paths is sometimes great, so a user may become overwhelmed with choice and consequently freeze up. Also, link markers may not always be obvious, and this can result in the user missing potentially valuable information.

Providing a graphical browser seems to be important in assisting users to gain some sense of their context, or their surrounding information space. Edwards and Hardman (1989) conclude that a 2- or 3-dimensional representation of the information structure is an appropriate navigational device. However, problems exist with such displays when the number of nodes and links increases and the structure becomes very complex. This illustrates the need for other tools and devices, both to assist in filtering the structure and to provide additional context.

Link Markers

An initial problem that many users have is in discovering where links exist. How are the links in a particular system distinguished to the users? And what is the extent of the link — that is, what part of the node is the link origin? For example, if some text in a node is linked to another node, how is it indicated to the user that a link exists, and secondly, how is it shown what part of the text is linked?



Note the greyed underline of the word 'buttons'. This indicates that the text is in the style 'grouped' and is linked. This picture show the result of following the link — the destination node is shown at the right in the window 'Help Goodies' and is the card with 'button' information on it.

Figure 2.6 *Link anchor, anchor extent, and destination using HyperCard grouped text as an example. The grayed underline of grouped text can indicate both the existence of a link and the link extent at the same time. This is useful if this convention is immediately recognised.*

In some systems there are standard conventions for this. Often some form of highlighting of the linked item occurs. A textual link might be in a different type style or a different colour for example. This is an easy solution that quickly indicates the link existence as well as its origin. But this might conflict with other forms of highlighting such as that used to add emphasis. Evenson and Rheinfrank (1989) suggest that typographers might design an entirely new set of characters that would signal 'hyperness', but which wouldn't detract from readability. Unfortunately this does not encompass other media such as graphics. Typographical styling is no help there, so another solution is needed.

Evenson and Rheinfrank go on to suggest, however, that some sort of new hypermedia design language using an action-based sign-set would complement the existing alphabetic symbols that are currently used. The new sign-set could be provided with guidelines for combining them in order to create visual cues that could result in showing not only that a link exists, but also the action that will result on following the link. That is, the link marker indicates the link origin and its type. The extent of the link might be indicated by an outline appearing when the link is selected. This has potential problems such as the initial extra overheads in understanding the types of links, but it does seem to be a promising solution for link marking as well as providing extra meanings to the visual cues.

Link Destinations

Just as in specifying what is the link origin, how is the scope of the link destination known? Does it relate to the whole destination node, or to just a part of it? This is a problem of the granularity of the system — how large are the 'chunks' of information that are connected? The smaller the chunks can be, then the more meaning can be taken from the linkages. If only links from one node to another are available, then the nodes must be made very small so that confusion as to the reason for linking is avoided. One way to do this is to have a multi-window system such as NoteCards where very small nodes are common. Another method is to use larger nodes but with the ability to link up smaller parts of the nodes with small parts of other nodes.

Backtracking and History

Another problem is how to return to a previously visited node. Both of these problems may be partially solved through the use of a rapid return or go-back facility. This is really a prerequisite in any computer system that encourages exploration — something may be attempted and if it's not what is desired, then it is easy to return to the previous state. In this case, a link may be followed and if it is decided that it's not relevant at this time, then it is possible to go back to the previous node. It provides a safety net that encourages exploratory learning [Carroll, 1982].

A supplement is some form of link-previewing or progressive disclosure, where some information about the destination will be provided before actually following a link. This may take many forms depending on the type of the destination node. For example, if the node were a graphic, then perhaps a reduced view of the graphic could be shown. If the destination were a text node, then perhaps the title and length of the node could be shown. If the node were an animation or video then a reduced version of it could be played, reduced both in time and in size. These forms of link previewing are beginning to be shown in some computer systems such as the Macintosh with the QuickTime extensions to its system software. It is only a matter of time before they begin to appear in hypertext systems.

Scope

Another problem associated with navigating systems is the problem of scope, or how much information there is. Should there be an indication of how many nodes and links are in this system, this web, or this document? Which navigational strategies should be employed if the extent or size of the system is unknown? If you are in the middle of following a path, for example, and you want to finish soon, how do you know how far there is to go to the end? Should you stop now and come back later, or are you near the end already? This indicates the need for some global status information.

Meta-information

The guided tour idea has also presented problems in navigation and in understanding. Because a guided tour is system-controlled, a user has, to some extent, 'lost' control of where they may go. This means that a user will often want to know why they have been led to a particular node. It seems that meta-information — information about the structure of the tour — is needed to help make the tour intelligible and to help the reader avoid disorientation.

Meta-information needs to be distinguished in some way to differentiate it from normal content. The use of different fonts, size and style, and spatial location can be used to indicate this.

Summary

In summary, problems involved in navigating hypertext result in user disorientation and manifest themselves in such questions as:

Where am I? Where can I go from here? What can I do here? Is there anything about <topic> here? How much is there about <topic> here? How do I get to <topic>?

These questions arise from the complexity of the structure of the system. In small systems, the problems can often be easily dealt with through the imposition of some structure, but in larger systems when the number of nodes and links is great, structural imposition is one part of a larger number of solutions. One solution is to provide tools that assist the user to gain a sense of the structure. These might include devices such as graphical browsers, maps and landmarks, as well as page-layout conventions such as headings and scope information. Other aids include:

— devices that can simplify the structure of the system such as paths, tours and filters;

— meta-information that provides explanations of conventions used in a node, as well as context information through explanations and points of view;

— backtracking facilities that provide a safety net for users who explore the system;

— search facilities that enable a user to bypass the structures of the system to locate information.

Navigational Aids in Current Systems

Introduction

This chapter analyses the tools that are used for navigating hypermedia systems and discusses their use and utility, with detailed examples from specific systems being presented. The main tools and techniques discussed are: Paths, Webs, and Tours; Maps; and Guides and Agents. Some of the problems that these tools overcome will be described as well as some of their limitations.

Paths, Webs, and Tours

Paths, webs, and tours are filtering and structuring devices that present a simplified view of the system to the user. A path and a tour are relatively passive in some ways, in that a default path is usually provided which has some specific context. A default next move is always provided at each node so that it can be navigated very simply. However, sometimes more than one link is available so other routes can be taken off the default path. A web is different in that the control of the route that is taken is solely under the user's control. At any time, the user must decide which link to follow. What the web does is present only those links that belong to the context of the web. Each node may have many links, but only those that belong to the web will be displayed at each node. So the web is a filtered part of the whole system that has been saved previously on some subject, and this subject provides the context for the nodes and their connections.

Intermedia Web

The Intermedia Web is a set of links stored in a database that connects a number of documents into a network structure. The web is basically a filtered set of links from the database that is saved as a separate web document. Documents may belong to any number of webs. Users may access a particular network of documents and links by opening a web. Each document will only display links that belong to that web, even though there might be many more links contained in that document. To navigate the web, users may select from the available link-markers in the original document. Another way is to use the Web View (described in the next section) which is the main orientation aid for navigation in Intermedia. It provides a graphical view of the web and all nodes are selectable so that navigation can be carried out using it.

The Intermedia Web is primarily a filtering device. It is used to provide a specific context for documents and links. Documents may be created using one of the Intermedia applications, while links are only created once an existing web is opened or a new one is created [Walter, 1989]. This means that webs provide a context for documents and links. Without this context, links would have less meaning. They would also proliferate within the documents resulting in too much choice for the link follower and consequently they would be of less use.

Path replay is seen as an attractive goal [Utting, 1989]. It is an idea similar to the web in that it is a saved group of links that can be followed. However, the path in this case is seen as linear and might be used by a professor to find an interesting trail through a system, to save this path, and to pass it on to their students. Then the students could replay the path without having all of the system to explore — thus reducing the time and effort expended in learning what the teacher (or guide) wanted to pass on. That is, the students wouldn't have to learn the details of how to navigate the whole system to follow the same path that the professor did. They would be able to just replay the same path using some simple interface mechanism which reduces the complexity of the system by simplifying the structure to a linear path, and by providing a simple and easy-to-use interface for following the path. Of course, at a later time the students may need to learn how to navigate the full system.

NoteCards TableTop and Guided Tour

NoteCards uses the metaphors of a TableTop and a Guided Tour to assist users in navigation and orientation [Trigg, 1988]. The guided tour facility is similar to tools such as Hammond and Allinson's (1988) guided tour idea, Bush's (1945) trails and Intermedia's Web view. In NoteCards it is a system-controlled navigation facility consisting of a graphic interface to a network of special TableTop cards. It is a specialisation of the browser card already described, where the user navigates by clicking on the nodes on the graphical map.

The TableTop is a special card designed to capture the particular arrangement of cards on the screen. Since NoteCards enables more than one card to be displayed at the same time, there is a problem for users in capturing the author's intended meaning, because that meaning may be obtained both through the cards' contents as well as through the spatial layout of the cards. The TableTop card was seen as a solution for this. It is a snapshot that records the list of cards, the shapes of the windows, their positions on the screen, the scrolled locations (vertically and horizontally) of the windows' contents, and the order in which to open the windows so that the original (possibly overlapping) arrangement can be preserved. [Trigg, 1988]

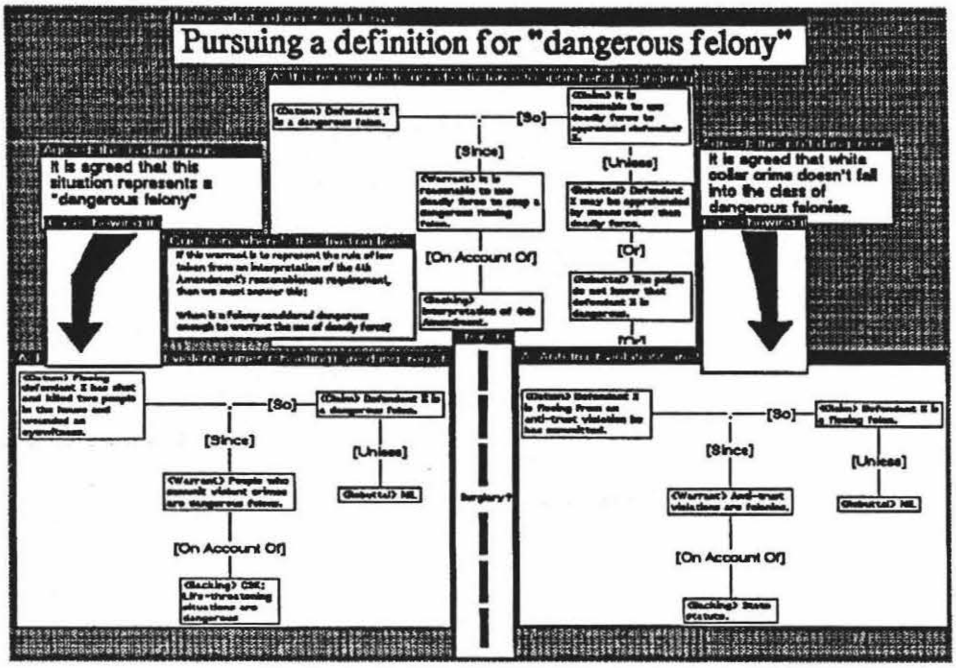


Figure 3.1 *NoteCards' TableTop card (from [Marshall & Irish, 1989]).*

There are, however, problems with the TableTop cards. One is in indicating the context of individual cards on the TableTop. In what order should the cards be read — is it left to right, top to bottom, or some other order? In an on-line demonstration pointing can be done by a demonstrator using a mouse. In an on-line guided tour the author must communicate in absentia. One way of enabling this is to open the individual cards on the TableTop in a specified order, so that the resulting simple animation provides an indication to the reader of the expected reading order. Other ideas have been to provide graphical arrows, asterisks and other devices to function as a stand-in for a narrator's gestures and emphasis [Marshall and Irish, 1989].

A guided tour card in NoteCards consists of a graphic interface to a network of TableTop cards. Specifically, a guided tour is a graph whose nodes are TableTop cards and whose edges are GuidedTour links connecting the cards. The TableTop cards may have other links connecting them to other parts of the network. The guided tour is accessed via a graph-based interactive interface allowing both authors and readers to work from the same concise overview of the guided tour's structure [Trigg, 1988]. Authors create guided tours using this graph-based card — TableTops can be linked with just a few mouseclicks and the resulting structure can be modified easily.

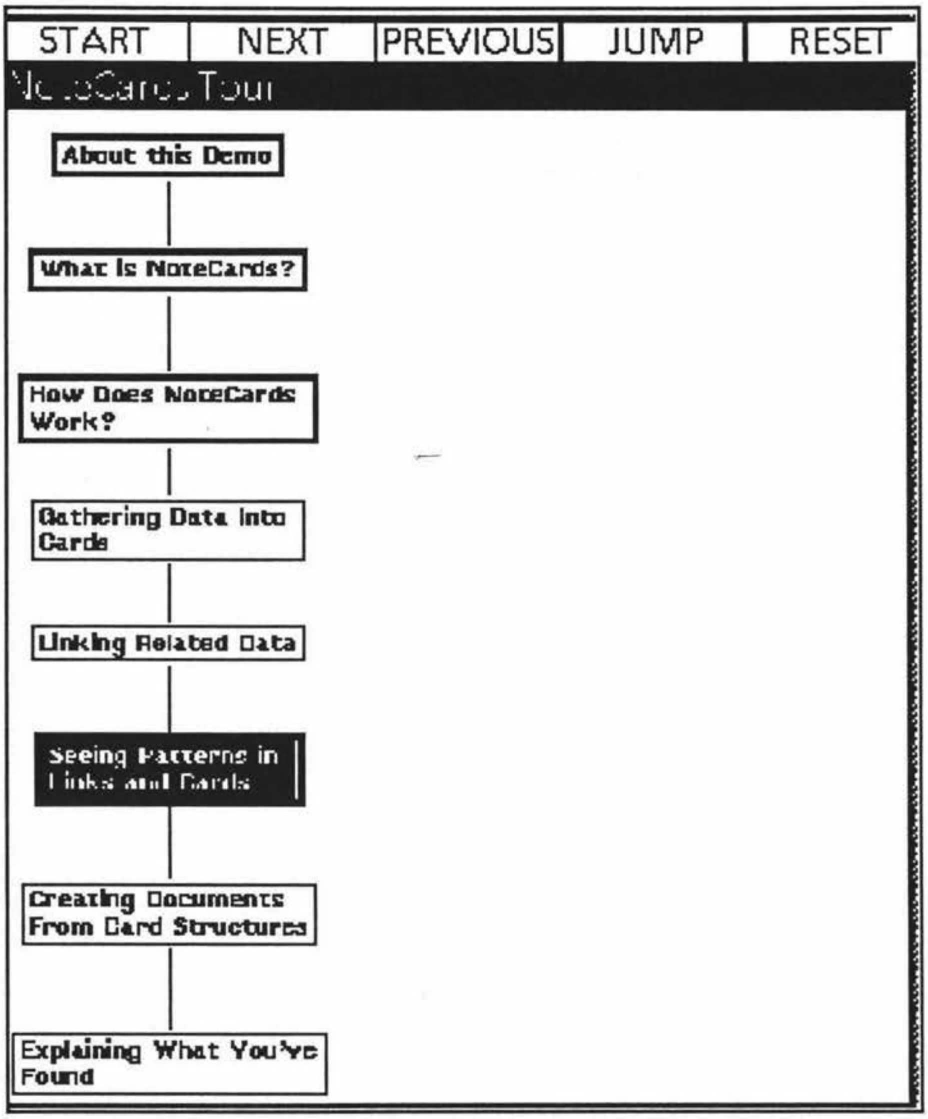


Figure 3.2 *A NoteCards Guided Tour card. This is a simple linear tour. Note the highlighting of the current node, the thick outlines around nodes already visited, and the navigational controls at the top.*

A guided tour in NoteCards can have multiple paths through it — it does not have to be linear. To use a guided tour, five commands are available: start, next, previous, jump and reset. When the Next command is issued and there are multiple destination cards, then a menu, listing the possible next TableTops, is shown with the user having to select one. The Previous command displays the previous nodes that were visited in the order that they were visited. The user selects one of the previous nodes and effectively jumps to it. A Jump command can be issued and then a node on the graph is selected to jump to. The Reset command closes the current TableTop, turns off any highlighting of nodes and links on the graph, and clears the list of previously visited nodes.

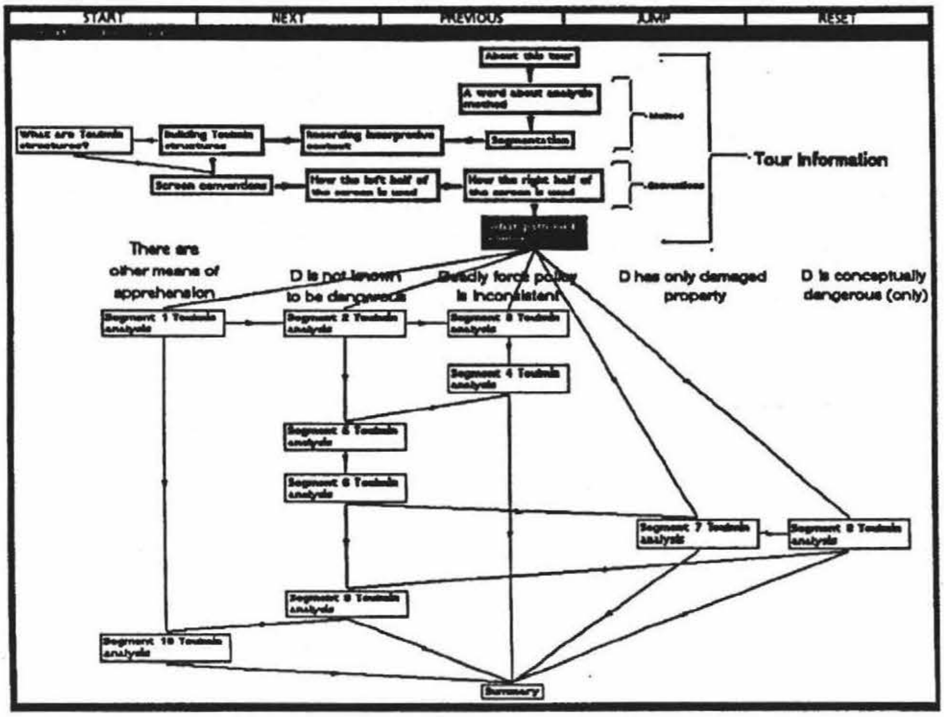


Figure 3.3 *A NoteCards Guided Tour Card with multiple paths. Note the graphical layout indicating the tour structure, the highlighting of the current position, and the navigational controls at the top of the card.*

The guided tour card has some special features that assist the user in reducing the load associated with navigation. The current node is highlighted on the graph — this allows the user to quickly ascertain where they are located in the path. Nodes that have already been visited are outlined with a thicker line than normal, as are links that have been followed. This provides a quick summary to the user of where they have been, how much of the path they have seen as well as how much they haven't yet seen.

A problem with NoteCards guided tours is in their ability to provide branching at any stage. Unfortunately there is no capability to provide a default path through the network and so the reader must decide which one to follow at the time. Also, no hints exist (apart from the title) as to what is contained in the options. There is no guidance from the author as to which path is more appropriate at that time. So this will add to the cognitive overheads of navigation as the reader must decide which branch to take as well as remember which branches might already have been followed.

Maps

A common tool for information access that has been taken from geography is the use of the map, of which there have been 4 main kinds — temporal maps, local maps, global maps, and fish-eye maps. Most maps use some sort of graphical system using small icons and lines to connect the icons. The icons often indicate the type of the node — so a text document will have a different icon from a graphic document, for example. The layout of the icons will indicate their relationship to each other in some manner. Spatial layout will give some indication of their relationship — those being closer together have more connection than those farther away, while it might also indicate a temporal relationship — a node close to the current node may have been visited more recently than one farther away. The icons will often be named to more readily identify their contents.

A map allows for direct navigation. Any of the icons on a map can be selected which results in the display of that node. This sometimes enables quicker access and it certainly allows for fast backtracking, which is an essential part of any system.

Temporal Maps

Temporal maps show the user interaction history, similar to the Hypercard Recent function. This is a graphical display containing miniatures of the 42 most recently visited nodes. If a node is visited more than once it is not added to the display, so this is not a true historical trace. Instead, it is relying on the user's visual memory to differentiate the appearance of each card. A better example is the history contained in the Intermedia Web View. This shows the path taken through the system up to the current node, with each node's name and the time that it was visited beside it. Document Examiner provides two types of historical records — a command history in addition to a path history. The command history allows users to redo any command previously used in a session [Walker, 1987].

Global Maps

Global maps attempt to show the whole information landscape. A global map will show all nodes and all links between nodes, so in one sense it shows all possible paths that a user may follow. A global map quickly becomes too complex as the number of nodes increases, however. In the end it has been deemed unusable [Utting and Yankelovich, 1989].

Local Maps

Local maps provide current context by showing the immediate area: nodes that have been recently visited and nodes that are accessible from the current node. In other words, a local map shows where you have been, where you currently are, and where you may go from here. Of course, local maps are quite dynamic. As different links are followed, the previous map that showed what nodes were accessible must change to reflect the new location. So the previous current node goes into the past, the previous accessible nodes, except for the current one, will disappear. Then the newly accessible nodes will appear. So each time a link is followed, the local map must change a great deal.

Sometimes local maps exist for a specific document and they help the user understand how the document is organised, so that they can have a better understanding of where they go within it. These maps generally are rather static, but they should automatically reflect changes to the document's contents as nodes are added and deleted. That is, the author should not have to make them or update them, they should be supported by the system.

Fisheye Maps

A fisheye map is a combination of global and local maps. It attempts to provide both local and global contexts through altering the map display depending on the Point of Interest (POI) [Furnas, 1986]. The point of interest might be defined as the current node, or in some cases it could be the cursor position. What the fisheye map does is display the immediate area in great detail, similar to the local map, but on the periphery it also displays some major details in order to provide wider context. So on the periphery icons representing multi-node documents might be seen, but in the POI area all the nodes within the current document and how they link up would be shown.

Map display could perhaps learn some lessons from geographical maps in terms of representation. Although different types of nodes have been represented differently on most maps, no indications of the node's size, interest, or other attributes seem to have been represented. This could perhaps be explored further.

For example, the size or amount of data contained in a node might be indicated by its icon size, font size, colour, or some other means. Also, if the map has been created as the result of a query, these characteristics might be part of the relevance feedback. As Frisse [1988] explains, the measure of a node's importance is not solely a function of the data contained within it, but also of the data contained in the nodes connected to it. There is the 'intrinsic importance' (the value of the data within a node) and the 'extrinsic importance' (the value of the data within the nodes connected to a node) of a node, and these might also be indicated through the use of different node attributes on a map.

An interesting idea for reducing the complexity of a map view of a network is the IBIS browser which enables the user to display or suppress the display of sub-networks or nodes and links [Begeman & Conklin, 1988]. Thus a sub-network of nodes is aggregated to form one node on the browser. This greatly simplifies the browser view and uses the dynamic abilities of the computer well. This might also be a way of providing a global map as described previously. A global map might display sites or aggregations of documents that might have been filtered as a result of a query. Then, by selecting a site, the local aggregates for that site might be displayed. Subsequent displays might result in a local display from which documents can be selected.

Intermedia Web View

The main navigational aid in Intermedia is the Web View facility which is a graphical browser that aims to give the user a sense of context or location within the web. Intermedia originally had three types of graphical browsers — a global map, a local map, and a local tracking map. The global map showed every node and the links between them. This became very large and complex as the number of nodes and links increased. The local map seemed more useful. It contained a map containing the current node and all the nodes connected to it. The local tracking map was found to be of most use. It was similar to the local map except that it was dynamic and would change to indicate the current state. Thus a user could leave the web view open and it would change its display as the user moved through the web.

However, it was decided [Utting, 1989] that only a single view of the web would be provided and a variant of the local tracking map was created. This was updated dynamically as the user travelled through the web and showed all the nodes the user could visit from the current node. However, it also included a path of the nodes already visited, so that the user could see where they had recently been, and a scope line that indicated how big the web was.

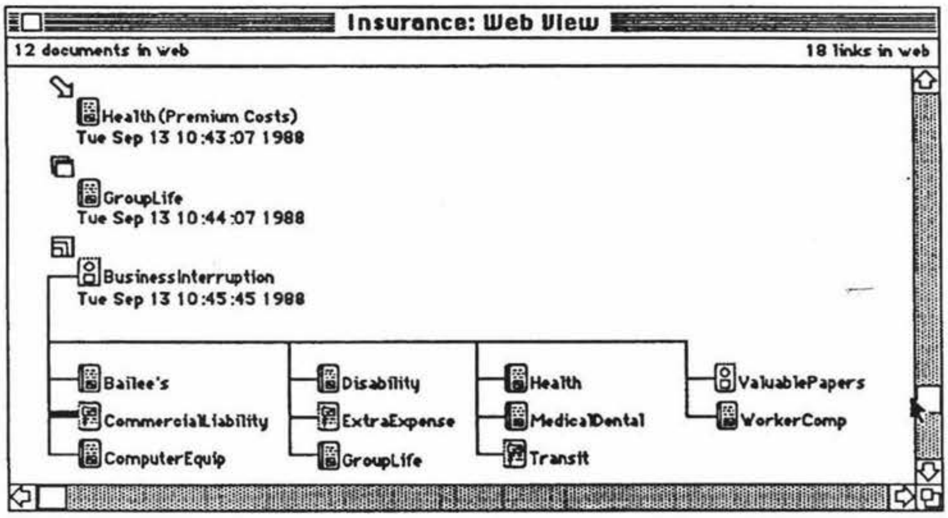


Figure 3.4 *The InterMedia Web View (from [Utting & Yankelovich, 1989]).*

The new web view provides a history and rapid backtrack facility through its path of nodes recently visited. All the icons on the web view are directly selectable so the user may select any of the already visited nodes and return to them immediately.

Link previewing is provided by the web view in Intermedia. It is the map of nodes that are connected to the current node. Each node icon displays information about the node itself including its name and type. When the user selects a link marker icon in the original document, the link that is selected is highlighted in the web view with a thick line. This indicates which node the link will lead to without actually having to carry the action out.

The Web View facility provides a scope line that shows the number of nodes and links in the current web. This gives a quick indication as to the global scope of the web which, when used in conjunction with other web facilities, gives a reasonable indication of where the current position is in the web, what path was followed to get there, how a previous node can be returned to, and how many nodes remain in the web.

Some other significant features of the web view are that it is unintrusive, flexible and efficient [Utting, 1989a]. Being unintrusive means that it does not require the user to help in any way in the creation of the map as some other systems require. The user should not have to spend time and thought manipulating the view of the web, they should be spending their time and thought on the web contents. The web view is there when and if they need it. Being flexible and efficient means that the view makes efficient use of valuable screen real estate. It will adapt itself to the amount of space that a user allots to it, allowing the rest of the screen to be used for the important task of viewing the other documents.

A number of enhancements have been suggested for the web view [Utting, 1989a]. The first is the issue of how long the path or history list should be. Some flexibility would be preferable but at the moment it is limited to the forty most recent nodes. The user should be able to specify how long the path should be. Another enhancement to the path would be to change the criterion for node inclusion, so that the user might include nodes by date or time, or perhaps by node type.

Another improvement would be to ensure that the web view showed the links at the block rather than document level, in order to ensure consistency of the interface. This was initially in the design of the web view. A user could select any document in the view and a series of hierarchical popup menus would display all the document blocks as well as the links associated with each block. This remains unimplemented at present.

NoteCards Browser and Guided Tour

NoteCards browser is an active structural overview diagram of the network of nodes and links. Each notecard is represented by its title in a box on the browser, and these may be selected to navigate the network. Different link types are indicated by different line patterns in the browser. The browser also enables the manipulation of the network so the structure can be modified. The browser is system supported so that as notecards and links are added and deleted, the browser card automatically updates the view of the network to reflect the changes.

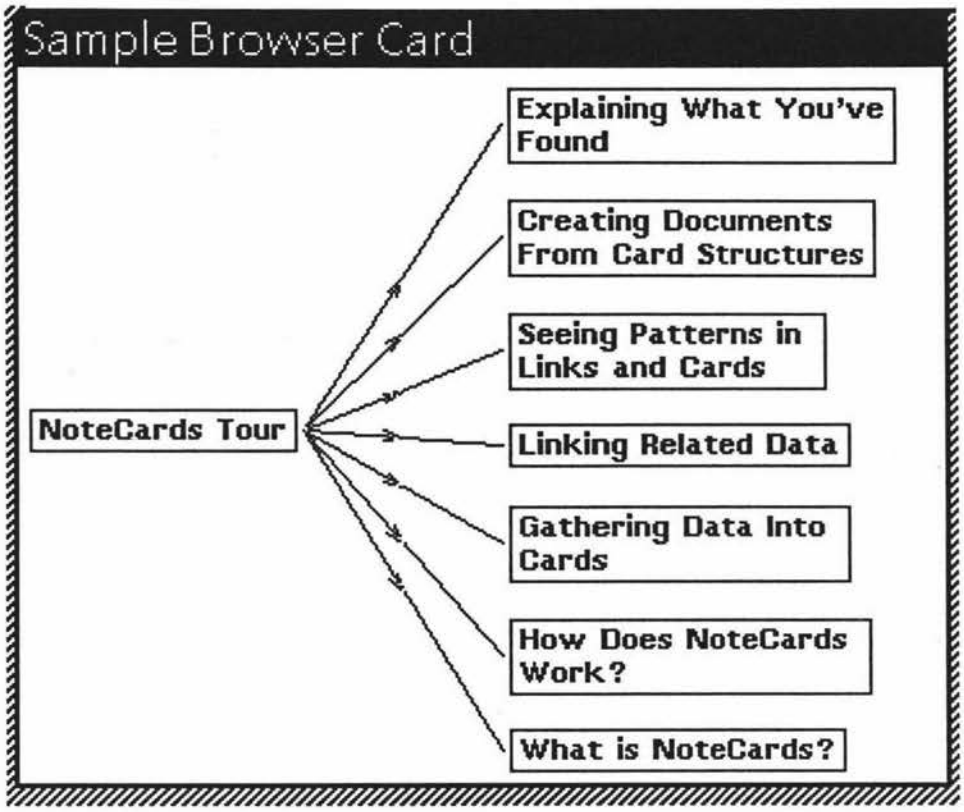


Figure 3.5*A simple NoteCards' Browser Card.*

Guides / Agents

Guides and agents have been suggested as entities that can provide:

— narrative, and integration through point of view [Laurel, Oren & Don, 1990] [Oren, Salomon, Kreitman & Don, 1990];

— dynamic personalisation of paths through the information space [Oren, 1990];

— delegation of routine tasks, filtering of information, intercommunication, and execution of complex tasks [Negroponte, 1990];

— entertainment, tutoring, information filtering, scheduling, reminding and advice [Laurel, 1990]; and

— a personal news service (Alan Kay in [Linderholm, 1992]).

There currently seems to be a separation of function between the two concepts of Guide and Agent. A guide is an entity that merely suggests things whereas an agent takes an active part in our interactions. Oren et al. (1990) define an agent as:

> "*an autonomous software entity that makes choices and executes actions on behalf of the user. They embody the expertise to find and present information to the user, responding dynamically to the user's changing goals, preferences, learning style, and knowledge.*"
>
> [Oren et al., 1990, p. 381]

This suggests that guides and agents may be more suited to differing user groups. A guide would be more suited to users who have little experience in some area and need assistance from a 'recognised expert'. An agent, on the other hand, might be more suited to expert users who have confidence that they can leave mundane chores to automatic agents. The agents would base their actions on the expert user's past actions which would show their interests and preferences. The agent would then automatically assess where to go in the information space depending on what the user had previously been interested in and what they appeared to be interested in at the current time. A guide is a more passive version of an agent. It may have some of the underlying intelligence of an agent but will merely suggest actions to be taken, rather than carrying them out autonomously.

One of the main ways a guide has been used is to provide a narrative point of view. A guide is given some characterisation so that the user may expect a certain point of view and so get context on the path that is followed. The guide can assist in a number of ways: providing commentary, suggesting navigational moves, and selecting content related to their point of view.

Assessing point of view is important whenever information is read, because the filtering and arrangement of information that occurs are due to the information provider's ideas of order, relatedness, causality, salience, relevance, and importance [Laurel, Oren & Don, 1990]. Presenting a guide as a character can help the user assess the presenter's point of view and so assess the relevance and context of the selected information.

Guides have been used in a number of projects. Laurel, Oren & Don (1990) describe issues involved in the use of agents and guides in the prototype multimedia database "The Americana Series: A CD-ROM Sampler of United States History" developed by Apple Computer in conjunction with Grolier, Inc. In this project, ten 'generic' characters of the period AD 1800 — 1850 were featured, each with obvious traits of gender, occupation, and costume. Point of view was derived from the degree of interest each guide was assigned for various topics in the database. After selecting a guide, at each node the cognitive overhead of navigation was reduced through the guide suggesting a next choice or favourite article. If this was not wanted then, by clicking on the guide icon, a ranked list of "Next moves" based on their degree of interests would appear, with the top of the list being the default move. If a different guide was chosen then a different set of next moves would be displayed.

Users often wanted to get more perspective on the guide [Oren, Salomon, Kreitman & Don, 1990]. They often wanted to know the guide's 'life story' and how it related to the choices that were made. They also wanted to know if the content of a story was from the guide's point of view or not. This was not the case — guides were only there to assist with navigation. An interesting aspect of their use of guides was the propensity of the users to imbue the simple characters represented with traits, such as emotional qualities, far beyond those actually represented. This could be a problem with the guide characterisation — human figures used as icons to represent a point of view might be better represented by a non-human symbol that offered similar contextual connotations. For example, the gold trail might be better represented by a pan and pick-axe than a gold-miner figure. Of course, some of the versatility and breadth of representation that a personality can offer might then be lost.

Summary

In this chapter some of the main navigational facilities available in some current hypertext systems were described. The main aids are:

— paths, webs and tours as exemplified by the Intermedia web and NoteCards' Tour;

— maps, as exemplified by the Intermedia Web View and other browser displays;

— guides and agents, as used in the prototype multimedia database 'The Americana Series: A CD-ROM Sampler of United States History' described by Laurel et al. (1990).

These aids attempt to alleviate some of the navigational problems presented by the complex structures present in a hypermedia system. Although they have been partially successful, additional aids, as well as refinements of the existing mechanisms, will be required to enable the multiple levels of hypermedia systems to be negotiated effectively by all levels of users.

Chapter Four

A HyperCard Path Facility

Introduction

This chapter contains a description of the Paths facility. First of all, the origin of the paths or trails concept is described, and some of its possible benefits are presented. Then there is a summary of common navigational techniques available in Hypercard, which was chosen as the testbed for the paths facility. Following this are some of the important concepts present in the facility. The ideas of following a path and exploring off the path are discussed in relation to the Hypercard environment. The provision of meta-information and its importance as an aid to navigation is discussed. Then the importance of feedback as an interface principle is discussed in relation to how it exists in the paths facility. Finally, some possible ways of using the paths facility are presented.

Origin of Paths Idea

The guided paths facility traces its history back to Vannevar Bush and his original ideas about enhancing intellectual productivity through the use of a device called the memex, a precursor to today's ideas about hypertext systems. He explains how it might work:

> *"The owner of the memex, let us say, is interested in the origin and properties of the bow and arrow. Specifically he is studying why the short Turkish bow was apparently superior to the English long bow in the skirmishes of the Crusades. He has dozens of possibly pertinent books and articles in his memex. First he runs through an encyclopaedia, finds an interesting but sketchy, article, leaves it projected. Next, in a history, he finds another pertinent item and ties the two together. Thus he goes, building a trail of many items. Occasionally he inserts a comment of his own, either linking it into the main trail or joining it by a side trail to particular item. When it becomes evident that the elastic properties of available materials has a great deal to do with the bow, he branches off on a side trail which takes him through textbooks on elasticity and tables of physical constants. He inserts a page of longhand analysis of his own. Thus he builds a trail of his interest through the maze of materials available to him.*
>
> *And his trails do not fade."*
>
> [Bush, 1945]

This illustrates some of the fundamental concepts behind the paths facility. Firstly, it illustrates that it is useful when there is a large amount of underlying data available. Secondly, connected information may be located in many different places and these connections may not be immediately apparent. Thirdly, the importance of annotations or meta-information is illustrated. Fourthly, the permanence of the path compared to human memory is established.

Underlying Data

The guided paths tool is useful in many situations, but more so when there is a large amount of underlying data. In situations when there is only a small amount of data, then the number of possible links is few and it might be possible to scan it in a reasonable amount of time. Also, the structure of the data will be important to its meaning. In situations where there is a large amount of data, however, there will probably not be an overall structure which adds meaning to the data. Rather, the data will be pools of seemingly disconnected material. Paths are a way of providing a structure over the top of the underlying data. They could provide many structures over the data — different structures may be appropriate for different groups of users.

McAleese states that:

> *"Structure imposed on what is browsed or on the process of browsing facilitates effective browsing. Users must have some knowledge of structure to build on."*
>
> [McAleese, 1989, p. 40]

This illustrates the importance of the structure that is created over the top of the underlying data. The meaning of a document or series of documents is not just a function of the data contained within it, the structure of the document contributes much of the meaning also.

Connections

These seemingly unconnected pools of information are certain to contain linkages between them, although the links may not be explicit. Guided paths can make the connections explicit. Existing documents can be linked in new ways to create new documents, new connections and, perhaps, discoveries for some users. The explicit structure produced when a path is created provides the base upon which understanding can be built.

Authors often cannot anticipate all the uses to which their material might be put, and so documents are normally arranged for a particular target audience [McKnight, Richardson & Dillon, 1989]. Paths enable an author to impose other structures over the material to satisfy the many different demands that users place on information due to their varying goals and expectations.

Annotations and Meta-information

When the user wants to record their own thoughts about the path they are creating then the ability to annotate the path is important. This enables the user to explain, for example, why they linked up two items or personal thoughts about an item — for example disagreeing or agreeing, providing emphasis — for example "pay particular attention to ...", and so on. This is important in providing context and to record particular thoughts at the time that they occur. See the section on meta-information for more details of the importance and uses of annotations.

Path Permanence

The path that is created is permanent and so it can be replayed at any time. This is quite different from human memory where previous lines of thought tend to fade over time. It's also different from normal system navigation, in that a particular path that is followed will be very difficult to replay because of the number of choices of possible links at each node. This permanence means that as time goes on, the original meaning that was obtained in a path may still be referred to. Of course, the permanency of the path is relative — it may be deleted or modified if so desired.

Nielsen [1990, p 189] hypothesises that the non-sequentiality of hypertext might have some social problems. He gives the example of a student arguing with a professor that material that was examined was not available in the assigned readings. The professor might claim that there was a hypertext link to the information in question. And that the students might counter-claim that the link was almost invisible and not likely to be found by a person who was not already an expert in the subject matter.

This would not be such a problem with the use of paths. Paths could be handed out that cover most of the required material with hints about what other links might be rewarding contained in the meta-information. Then the professor would have a better basis for examination with the main topics being on the path, while advanced topics are just off the path, but available for students who had the time or were motivated to explore further.

Learners differ, not only in terms of ability, strategy and temperament, but also in their goals and contexts. Learning is also supported by a vast range of activities, some active, some passive; some creative, some reactive; some directed, some exploratory. The nature of learning, and of the tools and situations that support it, is task dependent [Hammond, Mayes and Barden, 1989]. While the paths facility would be an appropriate tool for introducing students to a subject area, this style of passive learning is not appropriate for many learning situations. The developers of an Intermedia database gained more from their interaction with the system than did the students, who used it in a browsing fashion. This problem was recognised by the Intermedia designers and they have worked on tools to allow annotation and additions.

The paths facility can also be used in this manner. An assignment might require students to create their own path through the system on a particular topic. This would require more effort in navigation as well as in understanding the content in order that the links between nodes were relevant. This would therefore seem more appropriate for non-novice users. Currently the paths facility does not provide support for annotations when in student mode, only when authoring can extra information be added.

McAleese (1989) summarises that browsing and navigation are characterised by a number of concepts, some of which can be embodied in the paths facility. Excerpts from his summary that might be addressed by path mechanisms are:

> "• *Structure imposed on what is browsed or on the process of browsing facilitates effective browsing. Users must have some knowledge of structure to build on.*
>
> • *Browsing is a member of a set of information seeking activities or strategies best suited to covering a large and complex area without going into too much detail.*
>
> • *Browsing can facilitate discovery learning by providing the ideational scaffolding while allowing learners to find out for themselves.*
>
> • *Browsing requires personal and system filtering mechanisms to tailor the information presented to the needs of the user.*

> • *Typographic and iconic cues can be used to direct the attention of the user; and to draw distinctions between nodes and links of different types, eg ...*"
>
> [McAleese, 1989, pp. 40 - 41]

The paths mechanism can certainly address some of these basic concepts.

Sites, Modes and Trails Revisited

Nievergelt and Weydert [1980] were concerned with the users of a command language-based interactive system — not particularly with hypertext systems. However, the basis of their framework for the design of interactive dialogs corresponds somewhat with the concept of paths through an information space — in this case a path through a hypertext system, although it could be more generalised than that. They propose three concepts as the fundamental structuring tools for human-computer interaction — sites, modes, and trails.

Their concept of a site is a "*relatively small part of the data present in a system... Other data should be invisible at this moment, as it would only interfere with the 'active data'...*". This corresponds well with the idea of the nodes in a path being a small subset of the data available, that which is of particular interest to the user. The general information space consists of a large number of sites that may or may not be linked. Nievergelt and Weydert explain: "*A user moves around this space of sites, can see a map of parts of it, and can edit (modify) the space when he wishes to impose a new structure over his data.*" The concept of a site basically explains that, at any given time, a user is only going to be interested in a small subset of the information available, and that providing efficient access to that small subset is what is important.

Their concept of a mode is a situation where the actions possible are limited by the current situation. For example, when editing a picture, only graphics commands are of particular interest, and graphic commands should be the only ones available. Nievergelt and Weydert say that a collection of commands which is likely to be used simultaneously should be grouped into a mode. This can be seen in the paths facility, presented in this thesis, which provides a floating palette or a separate menu when a path is active. These provide all the possible commands available for paths. There are, however, two separate path modes — author and student. The basic commands available are the same, but the author has additional commands available for path creation and editing that are not available to a student and therefore are not presented.

When following a path, there are a number of options or commands available at each node. The primary ones are the path commands that are specific to the paths facility. But there also the embedded commands that are sometimes available at a hypertext node, and these may be followed on a side-trip, if so desired. However, the commands available are specific to the current node, site or situation.

Nievergelt and Weydert (1980) describe a trail as

"a feasible time sequence of pairs (current mode, current site) which describes a user dialog... Trails can be named, stored, edited, and invoked (re-used). "

[Nievergelt and Weydert, 1980, p. 332]

In a hypertext system then, a trail might be considered the linear sequence of node visitation over time. This is exactly what a path is. Paths can be saved, edited, and re-used.

These correspondences in function between the paths facility and a theory of interaction suggest other possible uses for a path, such as prototyping interface functionality or tutoring, for example. Its use in prototyping might only take the form of easily linking up nodes (or screens) into a meaningful order. This order could be easily edited to alter the interaction. The nodes could also be edited without necessarily affecting the order of interaction.

Hypercard and Navigation

Why Hypercard?

Hypercard was chosen as the test-bed for the paths facility for a number of reasons. Firstly, it is a very popular hypermedia system that was, until recently, bundled with every Apple Macintosh system. Thus it is widely available. Secondly it is relatively easy to use — it is widely used as a prototyping tool. Thirdly, because of its availability and ease of use, it has been widely used in the education sector to develop multimedia courseware or stackware. This means that a paths mechanism may become quite useful as libraries of stacks are built up.

For the most part, individuals have created individual stacks or small groups of stacks with few or no links to other related items. Now that Hypercard has been out for five years there is beginning to be a larger number of well-designed, well-produced stacks. Small libraries of stacks are beginning to be available as most stacks are shareware or in the public domain. This means that more people can have access to the information contained within them. As these libraries of stacks increase in size then there will be more possible connections between them, although these may not be explicit. The path tool can make more effective use of these resources by making it easier to incorporate parts of many different stacks to create new and unforeseen trails through the information available.

Hypercard's Navigation Facilities

Hypercard provides many intrinsic navigation facilities. Most of these can be found under the Go menu. A list of them and a short description of their functions follows in table 4.1.

HyperCard also contains other navigational mechanisms. A button or a field can contain a hypertalk script that can contain navigational instructions. Commonly used are buttons to let the user go to the next card, the previous card, the home stack, and to return to the card that they came from. These usually include some sort of visual effect, such as a scroll left or right, to provide feedback as to the function that is occurring.

Function	**Description**
Back	Go to the previous card. This can be a card in another stack if a jump was made to the current card.
Home	Go to the Home stack.
Help	Go to the Hypercard Help stack for information about Hypercard.
Recent	Bring up a visual map of the 20 most recent cards that have been visited in the order that they were visited. It is not strict temporal order — cards are not repeated on the display. Uses miniature bitmaps of the cards to aid recognition. Clicking on one of the miniatures leads to the corresponding card.
First	Go to the first card in this stack
Prev	Go to the card before the current card in this stack.
Next	Go to the card after the current card in this stack.
Last	Go to the last card in this stack.
Find...	Brings up the find dialog. Allows the user to search for a text string in a field.
Message	Brings up the message box and allows the user to enter a Hypertalk expression.
Scroll	Allows the user to reduce, enlarge, or scroll the current stack window.
Next Window	Takes the user to the next stack window, if more than one stack is open at the same time.

Table 4.1 *HyperCard's Go menu*

Other common functions include:

— *hot text* is text that, when clicked on with the mouse, will take the user to a card related to the text clicked on; and

— *invisible buttons* that may take the user elsewhere. These are commonly placed over graphic items and might provide further explanation.

This means that there is a large variety of possible mechanisms for navigating around a stack or group of stacks. It also means that some way must be found to simplify navigation for users. This might mean standard conventions, of which some de-facto ones exist already for basic navigation, or other means such as meta-information, that can provide an author's commentary and advice as to what to do or look for at a particular card.

Path and Off-the-Path Exploration

As with any document, a path might be seen as a complete document in its own right, but it can also be viewed as a starting point for further exploration. A distinction is made between following a path and off-the-path exploration.

Following a path

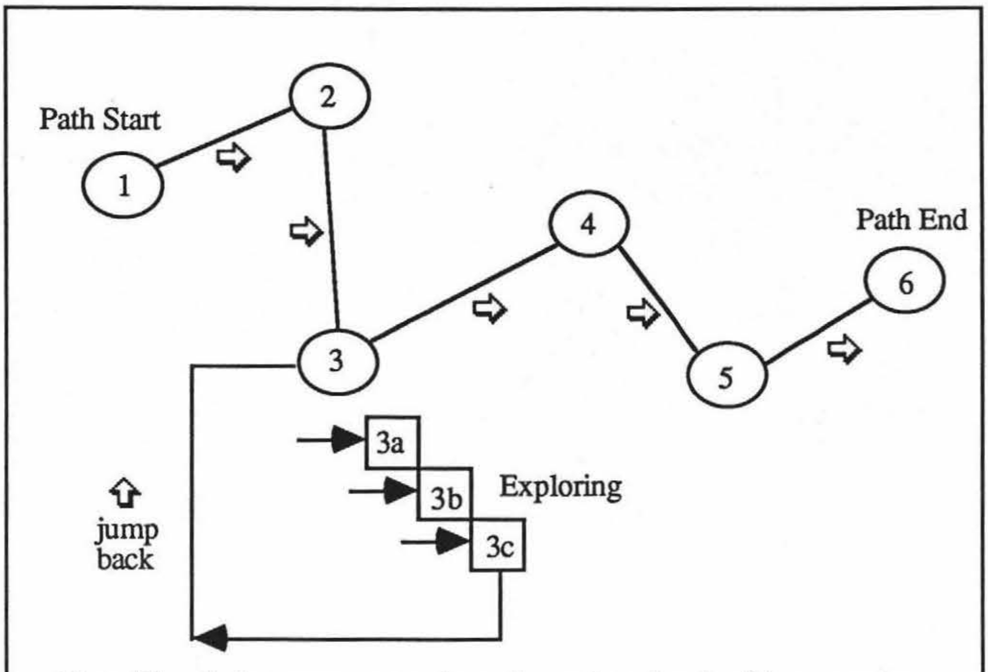
The path tool allows another overlying structure to be placed on top of the existing Hypercard structures (i.e. stacks). It provides extra flexibility in that pre-existing structures may be incorporated into new structures with little effort — it is not necessary to create buttons or scripts to build up a new path.

The newly created path is, of course, linear. However, anytime a number of cards are browsed, the result over time is linear. What the path does is allow any nodes to be visited in a particular order. The order that they are visited can affect the meaning contained within the cards — the links are not devoid of information, they affect the information content of the following nodes. Any particular card may be accessed as often as required — it just needs to be added in to the path structure at the appropriate position. Thus the effect may be as in a hierarchy — one card may be used as a landmark and returned to at particular times before going to another part of the system.

Exploring off the path

At any time while following a path, a user may also explore off the path. This enables a user to explore for more knowledge if they come across something interesting while they are following a path. It enables exploratory learning. That is, following a path has a certain reason behind it and is limited to what is on it, while exploring off the path enables the user to explore anywhere they like within the system.

The path gives the user some form of security against getting lost or disoriented while they are exploring off it. At any time, the user is only one mouse-click away from returning to the path at the point that they left it. So the user may explore freely, safe in the knowledge that they can quickly and easily return to a known location.



Note: The circles represent cards on the path and each of these can be located in any stack. The squares represent cards that are not on the path.

Figure 4.1 *Path and off-the-path exploration.*

Exploring off the path requires the use of other tools that can be classified into two groups — embedded facilities and HyperCard's own navigation facilities. HyperCard's navigation tools have been described previously, so in this section some types of embedded facilities will be described.

Embedded Facilities

Embedded navigational facilities are the navigational facilities that have been provided on a card by the stack author. These embedded facilities vary from stack to stack and even from card to card within a stack. So there is much variation which can be confusing. The main problem has been one of how to indicate a link marker. Different systems have different ways of solving this problem — some use small icons, some use colour or another form of highlighting to differentiate a linked item from a non-linked item. Unfortunately, Hypercard provides perhaps too much flexibility in linking and so a user does not always know where to click to follow a link. Some de facto standards have arisen, however.

These standards generally consist of generic icons that perform similar actions. The following defacto standards seem to exist:

	go to the next card
	go to the previous card
	go to the card that a hypertext jump was last made from (not necessarily the previous card)
	go to the home card
	go to the last card
	go to the first card
	display help

Table 4.2 *Some HyperCard navigational conventions*

This is good for basic navigation. However, potential problems arise when an author has built in more complex navigation tools such as 'hot text' or hidden buttons. 'Hot text' is a text item that is linked to another node. Hidden buttons are invisible buttons that are often placed over pictures to link to another node. Both of these types of links pose a problem for authors — how should the link marker be indicated?

Hot text might be indicated through the use of highlighting using another colour, typeface, or type style. Of course, then a user may get confused as to what indicates the author's emphasis and what constitutes a link marker. A small icon might be used although this might be distracting. Hypercard provides a text style called "Group" that can be used to treat a text string as one item. It can also be visually differentiated through the "Show Groups" command. This highlights all the grouped text items with a greyed underline that is unique to grouped text in Hypercard.

Invisible buttons pose more of a problem when trying to indicate a link. Even when a user is experienced, it is often very difficult to tell where invisible buttons might be placed and whether or not all of them have been selected. One method employed by experienced users is to hold down the Command and Option keys simultaneously. This provides a grey outline of all the buttons on a card and gives an indication where clicking might be productive. This is not always foolproof, however, especially when there are many buttons overlapping each other.

A path gets around the problems of using the embedded facilities through the provision of meta-information which can advise the user where or what to select. This is like having an experienced user on-hand and providing advice. But of course, once the path has been strayed from, this advice is no longer available and the user must find their own way about. This can sometimes be difficult and a user can easily get lost. Thus the provision of the safety net — the jump back to the path — is important for the user's confidence. They are always able to get back to a known location.

Meta-information

Meta-information is information that is supplementary to the content of a hypertext system. In the context of a guided path, it is additional information supplied by the author designed to make the path intelligible and to help the reader avoid disorientation. It can preview the contents of a path, explain conventions used in the path, and provide reasons why the path exists and what the author's perspective was when creating it. Other research has suggested the need for introductory information about content and structure in databases, rather than just information about search principles [Linde, 1989]. This would help database users get a better idea of context and enable them to carry out their task using a better mental model of the information available.

Marshall and Irish [1989], in referring to guided tours in NoteCards, identify four types of meta-information that are needed:

— expository text referring to the original network;

— instructions to the reader on how to interpret the screen layout;

— description of the structure of the tour; and

— textual and annotative devices that offset the effects of fragmentation.

Some of these are more applicable to the NoteCards environment than to Hypercard. More specifically, fragmentation occurs in NoteCards because NoteCards is designed as a multi-window environment. Thus nodes are usually very small, and multiple nodes might be used for presenting one idea. So one screen may contain many windows and some type of annotative devices may be needed to explain the layout to the reader. Otherwise the cognitive overheads increase as the reader struggles to understand how windows relate to each other rather than concentrating on the contents.

Expository text at the beginning of a path should discuss why it was created, what particular audience it is aimed at, if any, and what the goals of the path are. It might also provide information about the path structure. This will help the reader anticipate the content of the path and ascertain its relevance to them.

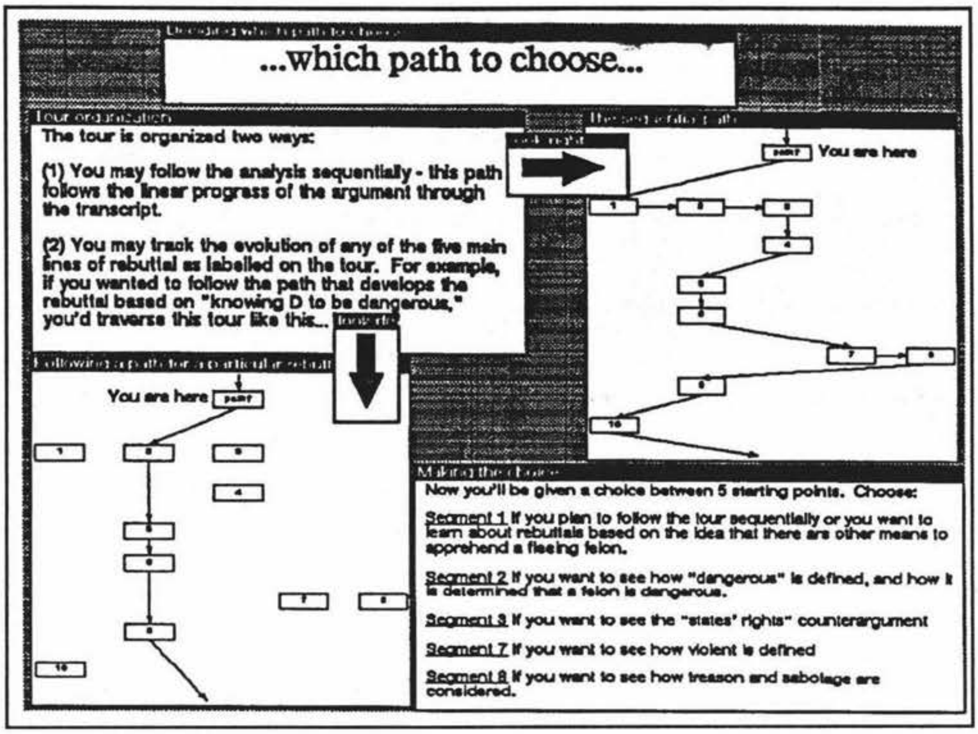


Figure 4.2 *NoteCards meta-information devices. A TableTop card used in a Guided Tour. Note the description about how this tour is organised and the arrows used to direct attention (from [Marshall & Irish, 1989]).*

Some information about the screen layout may be needed, especially in the Hypercard path environment. When nodes are taken from many different stacks and no particular standards exist, then some explanation might be needed about screen design at each node. If there are some standards, then these should be explained at the beginning, again to help the reader reduce their cognitive overheads and to focus on content.

Meta-information should also be available for the links from node to node so that if a user wonders why this link has been followed then some additional information explaining the link might be available. Of course it might not be available on purpose if one of the aims is for the student to form their own conclusions — perhaps being more creative in the process.

Feedback

Visual effects are used in two main places to provide feedback about the action that is occurring. Firstly, when a the mouse is held over the palette and the mouse button is depressed, the button underneath the mouse pointer will highlight so the user knows which button is being chosen. If the mouse pointer leaves the button area then the button will unhighlight. If a button beside the first button is now underneath the pointer, then it will become highlighted. This provides fast effective feedback as to which button the user is choosing.

Visual effects are used to provide feedback in describing the transitions from node to node. In a normal transition from one node to another (forward or backward) a visual wipe left or right is used. This simulates a page turning in a book. In a jump back to the current node after a side path has been followed, an iris close effect has been used to indicate the jump back to the current node of the path. Choosing the information button results in another window appearing. It is movable, resizeable and able to be closed. This window contains meta-information if the author has decided to provide any. The other buttons are only available in the authoring component of the paths facility, and are not explained here.

The consistent use of these feedback mechanisms can help the reader know which button they are choosing and to understand the function of the transitory buttons on the path palette. The transitory buttons are those which result in a change of position for the user — that is, when the user goes to another location. Feedback provides the reinforcement that the function they chose is actually occurring.

What Can the Paths Mechanism be Used For?

The paths mechanism can be used in a number of ways. Some uses for such a facility are as a guided introduction to a topic, a tutoring facility, a contextual reminder, a path re-finding tool, and as a bookmark and annotation tool.

Guided Introductions

Using the paths facility, guided introductions to topics can be easily and quickly created. This implies that there is a path author who is an expert on the subject matter. The author can easily browse through a large number of stacks and save interesting cards to the path, as well as add annotations and meta-information to assist the path-followers. Thus an introduction to a subject area may be rapidly built using pre-existing materials, with the only effort required on the part of the author being the selection and structuring of the path contents, and the addition of meta-information.

Tutoring Facility

Using the paths facility as a tutoring tool differs from using it for guided introduction mainly in the amount and type of meta-information supplied, as well as the type of cards selected. The meta-information might include other directives in this case, such as *"Imagine you were in this situation. What actions would you need to do in order to recover quickly?"*, or *"See page 123 of your textbook. Attempt numbers 1 - 5."*. This would provide a different effect overall than the guided introduction, but could again be useful.

Of course, in a tutoring situation, there is nothing to stop the tutor from creating their own stack of question cards and then using the paths facility to link each of the question cards in at appropriate points in the path.

Contextual Reminders and Path Re-Finding

The paths facility can be used in providing context, especially in situations where a path is being followed through a group of materials and some interruption occurs. When the path is returned to, the rapid scan through of the path up to the current point may assist the path follower to gain a sense of their previous train of thought.

The history facility included as part of the paths facility automatically records all cards that have been visited, so if an interruption does occur, the history may be saved, and then later converted to a path for use as a contextual reminder.

A similar use is the re-finding of important information. This is often difficult in large systems, when it is known that an item of information is there, but its exact location is unknown. A rapid scan through some related paths may assist us in locating the item, as the paths may remind us of the context in which the original item was located.

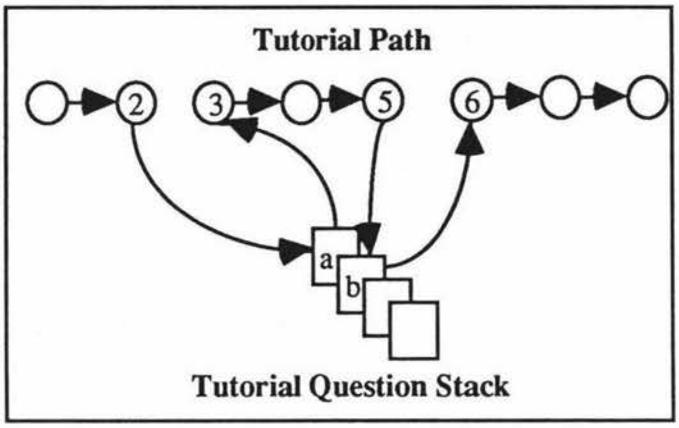


Figure 4.3 *A tutorial stack used as part of a tutorial path. Note that after card 2 on the tutorial path, the user is taken to card **a** of the Tutorial Question Stack. After that, the user is taken to card 3 of the tutorial path. Again, after card 5 of the tutorial path, the user is taken to card **b** of the Tutorial Question stack, and then back to the tutorial path at card **6**.*

Bookmark and Annotation Device

Similarly to the idea of regaining context is the notion of the electronic bookmark to save our place for later access. In this case, we might decide just to save an individual card that is important, and perhaps annotate our own thoughts to the card to explain its importance. Thus we might have a number of such bookmarks, either individually or as a series linked in a path.

Summary

The HyperCard Path facility developed in this thesis is a type of virtual structure. It provides an additional layer over the top of the existing structures of stacks and cards, and so provides extra flexibility in accessing cards from multiple stacks in specific orders.

The problems involved in accessing information contained in large numbers of different stacks poses problems for many users, and the provision of a path facility can reduce some of these problems by providing a simple and consistent navigational interface, and through the provision of meta-information.

The potential value of this paths facility increases as the number of stacks available increase, because it provides a number of ways to lessen the navigational problems associated with finding information in large systems. Some of the specific uses for the paths facility are as a guided introduction to a subject area, as a tutorial device, as a context-reminding tool, and as a bookmark and annotation device. These might be used to assist personal and group research, teaching, and learning.

Path Making Facility: Technical Details

Introduction

This chapter contains details about the design of the paths facility. The structures containing the path's information and the reasons behind design decisions are offered. Problems encountered during the design, and the implemented solutions, are discussed. The use of the History list is discussed and it's use in creating new paths is described. Problems concerning HyperCard's limitations and how they affect the paths facility are described. Finally, the problem of maintaining path integrity, and the lack of this feature in the paths facility, is discussed.

Path Structures

Each path is stored on one HyperCard card. The name of the card is the name of the path. There are three fields on each card — one contains the path information, one contains the meta-information and one contains the history list.

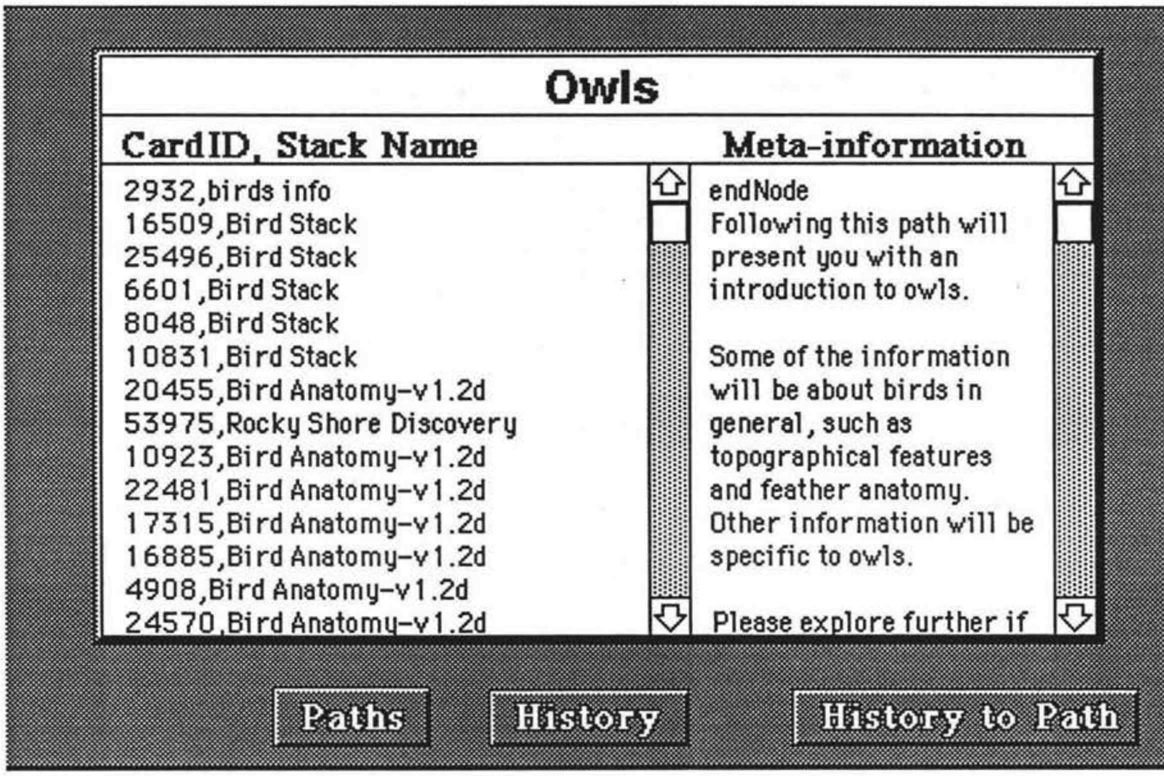


Figure 5.1 *Path card showing the path information and the meta-information*

Path Data

In the paths facility, the following data is required in order to navigate the system.

- *Card identification*
- *Stack name*

This information is stored on one line of the field. Each line represents one node and many lines of this information make up the path. The links can be from one card in a stack to any other card in any other stack. Each card will be uniquely identified by its ID number (which is produced by HyperCard itself) within a stack. Using the name of the card was considered but this was rejected due to the fact that it would not be unique to a stack, and it also required the stack authors to name the cards which does not always happen. So in order to be unintrusive, the card identification number was used to uniquely identify a card in a stack. Using the card identification number is also much faster in searching than using the card name. Of course, the stack name is also required to uniquely identify the card within the system (all the stacks available). In a networked version, some location information would also need to be stored.

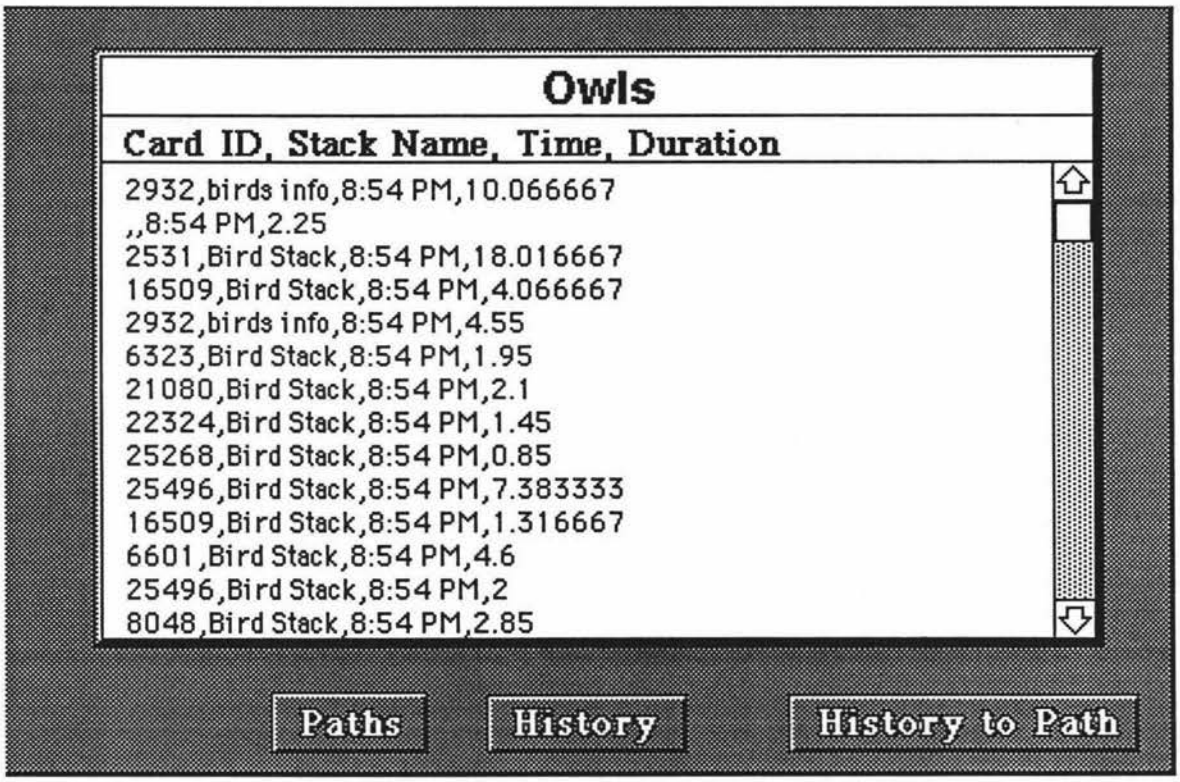


Figure 5.2 *Path card showing the History List.*

Meta-information Structure

The meta-information is stored in a separate field and is structured like this:

Delimiter
Meta-information for node 1
Delimiter
Meta-information for node 2

Delimiter

...

Meta-information for node n
Delimiter

Delimiters are on separate lines by themselves. The delimiter that has been used is '*EndNode*'. Using a whole word rather than a specific character was implemented because any character might be used in the meta-information itself, so using a whole word reduced the possibility of confusing the delimiter with the meta-information. Thus the meta-information may contain blank lines and any characters at all — the only restriction is that it may not contain the delimiter on a separate line by itself, although the delimiter may be used within a line and will then not be recognised as such.

History List

When a user is exploring, their path history is automatically stored. This history is saved only when the path is saved. Their path history is their linear trail through the system over time. So that includes nodes that were visited off the path as well as ones that were visited on it. The history list can be used to navigate also — any of the nodes on the list are selectable so a node can be re-visited just by selecting one of the lines in the list.

The data stored on each line of the history list is as follows:

Card identification

Stack name

Time of visit

Length of visit

As in the path field, the card identification number and the stack name are required in order to identify the card. The time of visit is stored for contextual purposes — perhaps to see when a particular card was last visited. The length of the visit is the amount of time in seconds that a particular card was viewed.

The length of visit might be used in a number of ways. Firstly as a contextual clue — it can be seen whether or not a card was examined in detail and perhaps gain a clue as to its interest level. Secondly, this length of visit data could be used as a criterion in creating a new path from the history data. As a basic assumption, it could be said that the length of time that a card is attended to corresponds in some way to its interest level. Thus, one quick and easy way of creating a new path would be to filter the history list using the length of visit data as the criterion.

Using the history list to quickly create a path has been implemented. On each path card there is the path field, the meta-information field and the history field. There is also a button called '*History to Path*'. Clicking on this button will enable a new path card to be created using the current card's history list as a base.

Creating a New Path from the History List

To create a new path from the History List, select the '*History to Path*' button on the path card whose history list is going to be used. The threshold value will then be asked for. The *threshold value* is the cutoff value (in seconds) for adding a card from the history list to the new path. With each history is the length of time that a particular card has been visited and if this is greater than or equal to the Threshold value then it will be added to the new path.

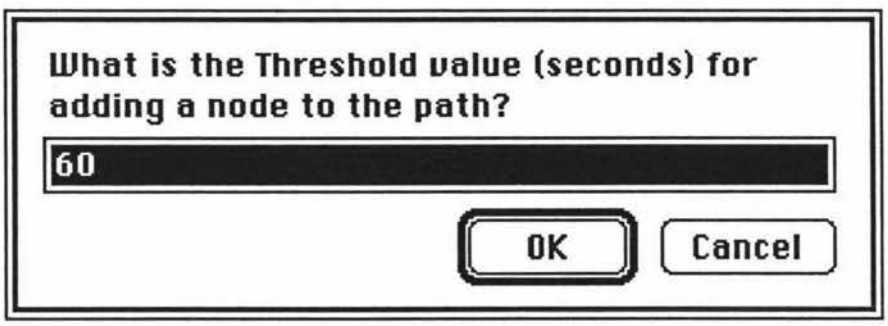


Figure 5.3 *Enter the threshold value dialog*

When all the cards on the history list have been examined, if some of them have been added to the new path, then a name will be required for the new path. A name should be entered describing the new path (as in Figure 3 in Appendix "The Paths Facility - How to Use it") or it can be named later — it is the name of the card, and a new card will be created with the path field filled with the cards satisfying the threshold criterion. The meta-information field will also be ready to accept new meta-information.

Problems Encountered & Solutions

While it would have been possible to store all the information on one field, it is a better breakdown by functionality to store the meta-information separate from the path information. The meta-information, while important, is a separate entity from the path itself. Having it separate also assists in lessening the effects of HyperCard's limitations. More specifically, HyperCard limits the amount of text in a field to 30000 characters. So the length of a path, as well as the amount of meta-information, is limited by this. The meta-information is more likely to come up against this limitation.

The history list might also come up against this limitation. If a user interacts with the system for a long time and covers many nodes then the history list will get very large. For example, assume each node in the history list takes up 50 characters:

card id	— 5 chars
stack name	— 25 chars
time	— 10 chars (max.)
length of visit	— 10 chars (max.)

This means that a maximum of 640 nodes may be added to the history list. Of course this is highly dependent on the stacks that are visited — if the average stack name was only 15 characters long then the number of nodes that could be visited before reaching the limit would be 800. Another way of reducing the size of the history list would be not to store the time of the visit. This does not seem to be particularly helpful unless a user comes back at a later time and can see when a node was last accessed [Utting, 1989a]. Perhaps this information should be stored in the path itself and could then be displayed on the path palette.

Storing the meta-information separate from the path information does complicate processing somewhat, especially when modifying a path, as it results in two areas that need to be searched before modification can occur. In a different implementation the information for each node might be stored together to ease processing and make it more functional. By storing all the information for a node on a path together, modification of the path would become easier. This would make a graphical tool to manipulate the path simpler to develop. It would also enable the extension of the path tool to encompass more types of documents rather than just HyperCard stacks and cards. Conceivably the path tool could be made generic and different types of documents (for example, graphics, movies, text, sounds) could be linked. Then each document could be opened with its own application with the meta-information being available as a side-note in a separate window.

This might be accomplished by creating a 'PATH' document that contains resources describing all the necessary details, such as document locations and names, which part of each document is linked, what application is required to display each document, who created the path and on what date, and so on. Of course, it would also contain the meta-information for each node.

Of course, this is what I have done but instead of creating a 'PATH' document, I have a card, and on the card I have the structures containing most of the information required. At the moment, destinations are limited to cards alone, no smaller destination chunks are available, so that linking from one field to another is not immediately accomplished. But, through the provision of appropriate meta-information, the reader's attention can be directed to the most interesting items.

Each application and document might be opened and manipulated through the use of Apple Event calls. These are commands that may be passed from one application to another and actioned on if the application recognises the command, without the user having to actually specify the command.

Integrity

There is currently no control over the deletion of cards and stacks or the movement of stacks to different locations. There is no way of ensuring that a node in a path is actually there. If a stack is moved to a folder that is not in the HyperCard search path then the stack will not be found and the user must locate it in the system. Similarly, if a card is deleted then all the paths that it belongs to will not be updated to reflect the change. What happens is that when the non-existent card becomes the current card in the path, the system will report that the particular card does not exist any more, and the next card in the path will be accessed. This can be a problem for novice users if they are not familiar with the Macintosh file system. If they are asked to find a stack then they may not know what to do. This needs some sort of attention and probably would be best addressed at the system level in a similar manner to the IRIS Hypermedia Services. That is where the system itself provides basic support for nodes and linking and manages them relatively seamlessly, so that the integrity of the system components is maintained without additional effort being required from users of the system.

Summary

The essential details required to implement a paths facility are:

- Node identification
- Node Location

These can be stored in a linear structure with the links being implicit in the order of the node information. In order to more fully implement a paths facility that provides annotation capabilities, some provision must be made for the addition of meta-information at each node. Meta-information may also be required for the whole path, and perhaps for each link also.

The provision of a history list is valuable, both for the immediate purpose of backtracking, when explorations have been made off the path, and also for the creation of new paths using some criterion, such as time, to select interesting nodes to add to the path. This means that more information must be stored in a *History* structure which might contain the node identification, node location, and the amount of time spent at the node.

If branching paths are desired then more complex structures will be required. This would include, for each node, a list of possible 'next nodes', with the first on the list perhaps being the default next node. Thus the user could either choose the next node from a list or just follow the default path through the information space.

For a paths facility to be most effective, the full range of hypermedia services need to be integrated into the operating system in order to fully support object linking and to ensure integrity of the paths.

Chapter Six

Pilot Study

Introduction

This chapter presents a description of a pilot study that was carried out in order to test the usability and potential effectiveness of a paths facility in aiding navigation. The experimental setup and results are presented, and these are examined to identify any significant features that require further study, especially differences between experienced and novice users. The apparent importance of meta-information (of which scope information is one part) is described, and the possible value of the paths mechanism is discussed.

Aims of this Study

The aims of this study were to investigate how useful a path facility would be for navigating hypertext systems, and to see if there were noticeable differences between different classes of users, specifically between experienced and novice users. These differences might be in the user's understanding of the system structure and operation, their actual use of the facilities provided in the path as well as the embedded facilities in the stacks, and in the amount of relevant information covered during a directed task.

Some of the questions that were considered during this study were:

- Would a path mechanism help users navigate a large hypertext system?
- Would novice users find a path mechanism more helpful than experienced users?
- Are there observable differences between groups on experience versus correctness?
- Are there observable differences between groups on usability over correctness?
- Can other possible trends be identified?
- Are there other interesting features of use?

Method

Twelve subjects were assigned to two groups with six in each group. Each group had the same information base to explore. One group used a home card as a base for navigation while the other group used the path facility. Their task was to find out what they could about owls, as well as general bird characteristics, specifically feather anatomy, surface characteristics, and flight. The path group had a path about Owls and bird characteristics made up for them while the Home group had to use HyperCard's facilities as well as the embedded facilities in each stack to find what they could. Materials introducing navigation in HyperCard as well as the concept of a guided path were given to the subjects for study beforehand. These can be seen in the Appendices. The Path group could, of course, also use HyperCard's facilities and the embedded navigational controls in each stack, if they wished.

The information base consisted of six stacks that were 'real-life' stacks. That is, they were not specifically created for this pilot study and were selected for providing some degree of realism in interface and content differentiation. They had varying degrees of value in relation to the task that was set. A breakdown of their contents can be seen in Table 6.1.

Stack Name	Number of Cards
Aesop's Fables	348
Animals	3
Bird Anatomy	47
Bird Stack	45
Dinosaurs	42
Rocky Shore Discovery	137

Table 6.1 *Contents of the Information Base*

The Owls guided path consisted of 14 cards selected from two of the stacks. The path contents can be seen in Section 12 of the Appendices. Note that the nodes on the path did not contain all the information that was required to correctly answer the comprehension questionnaire.

Each group had 30 minutes to explore the system and find out as much as they could. No note taking was allowed. At the end of the 30 minutes, all the reference sheets were collected in and questionnaires were handed out. The paths group had to fill out a subjective evaluation on the usability of the path, including questions about path navigation and their understanding of the palette functions. Both groups filled out a subjective evaluation on the usability of the system — mainly about HyperCard's navigational facilities and the embedded facilities in the stacks. Both groups also answered a multiple choice comprehension questionnaire on the task — on the subjects of Owls, feather anatomy, flight and other characteristics. This was aimed at testing the amount of relevant material that was covered.

Subjects were also encouraged to provide additional comments about the system that they used, and these were written at the end of each questionnaire.

Results

In collating the results, some of the aims are:

1. See if there is any perceived difference between the groups when comparing experience of use vs correctness in comprehension.

2. See differences in subjective evaluations of paths and overall between experienced and novice, and over groups.

3. Identify trends, if possible.

4. Pick out interesting features.

Comprehension results

There were 13 questions in the comprehension exercise. Scores ranged from 2 (1 novice, Home group) to 12 (2 subjects: 1 novice and 1 expert, both in the path group). Ranges within groups were:

	Novice	Expert
Home	2	11
Path	5	12

Table 6.2. *Range of Comprehension Scores*

Experience versus comprehension

Overall, the scores for the group that followed the path are higher than the group that used the home stack. This is not particularly interesting as that is what was expected, since the comprehension questions were aimed at the material that should be covered in the path.

However, previous computing experience seemed to have a greater impact on the scores of the Home group than it did on the Path group. There seemed to be a trend in experience versus correctness for the Home group. Users with greater computing experience did better than users with less computing experience. This was also the case for the Path group but their results in the comprehension questionnaire seemed to be less affected by their previous computing experience. In fact, one very inexperienced subject did as well as anybody on the comprehension exercise.

One subject in the home group had little computing experience but did very well in the comprehension questions, but this can be explained by their previous knowledge on the topics of owls and bird physiology.

Usability Questionnaire

Subjective Evaluation of System Usability

The subjective usability questionnaire aimed to gauge the users' feelings about the operation of the system and their confidence in navigating the system. It was also intended to provide a comparison between how a subject felt about their use of the system and how they scored on the comprehension exercise.

Feeling Lost

There didn't seem to be much difference in feelings of being 'lost' in the system. Expert users in both groups expressed fewer feelings of being lost than did novice users. Novice users in the path group still felt lost to begin with, but most indicated that after some experimentation at the start, they didn't feel lost from then on. The feelings of being lost were reflected in the comprehension scores — the subjects who felt more lost scored lower than those who didn't feel so lost.

All relevant information found

This question resulted in great confidence in the path group that they had seen all relevant information. Four out of six said they had definitely seen all there was in the system about Owls. This was usually reflected in the comprehension score although one novice subject scored poorly after expressing confidence that they had seen all the relevant information.

Subjects in the Home group expressed less confidence that they had found all the relevant information than those in the path group. Novices had less confidence than experienced users, as the expert users tended to exhaustively search the system for relevant information.

Confusion

Subjects in the Path group were less confused with what to do next. Again, the experienced subjects in both groups were less confused than the novices, but the between group comparison indicates that the Path group overall had fewer problems in deciding what to do next.

Card Navigational Facilities

There didn't seem to be a difference between the groups in understanding card navigational facilities, but there was between experienced and novice users. Novice users had more difficulty understanding the navigational facilities on each card.

Information

In both groups novice users were more likely to want extra information about what they could do on a card. There didn't seem to be any particular difference between the groups.

Confusing Controls

Subjects in the Path group found the controls more confusing than subjects in the Home group. In the Home group, experts found the controls less confusing than novices, while this was not noticeable in the Paths group.

Interaction Style

Half the subjects, evenly distributed between experienced and novice users, rated voice interaction as their preferred interaction style. Whether this is due to the novelty of voice interaction is unclear. Most users who knew what a floating palette was stated that it was good to use as it presented a consistent set of tools (to paraphrase one of the subjects' comments).

Novice users in the Home group didn't know what a palette was and so couldn't evaluate it. They usually rated menus and HyperCard's facilities as preferred interaction facilities, but this can probably be explained by their lack of knowledge about other interaction styles.

Interestingly, only one user (an expert in the Home group) thought that commands typed at a keyboard would be reasonable to use, although his preferred style was voice interaction, and another preference was a floating palette.

Other interaction styles that were suggested by users were gestural commands and a light pen. An experienced subject in the Path group suggested a gestural interface. Gestures could easily indicate 'Go to Next in Path' and 'Go to Previous in Path' with gestures right and left, for example. A jump back to the path might be indicated by a gesture up. Indicating the need for meta-information might be more problematic.

A novice user in the Home group suggested a light pen interface, and this might be because of their desire to find out where invisible buttons were quickly and easily. The user was not unfamiliar with using a mouse, so a more direct method of interaction, such as a light pen or touch screen, might be useful for some users.

Path Usability

The path usability questionnaire was designed to get the users' subjective evaluations of the usefulness of the path facility, an idea of how they used it, and whether they understood the path palette controls. It was also intended to provide some comparison between how a subject felt about using the path and how they scored on the comprehension exercise.

Path Coverage

All except one expert subject covered the whole path. The exception was a subject who felt more confident in his own ability to find the relevant information than in the path's ability to guide him.

Meta-information

All subjects, except for the one expert subject mentioned above, used the meta-information button on the path palette. Some used it all the time, while others only used it occasionally. There didn't seem to be a marked difference between expert and novice usage. However, novice users found the meta-information to be very useful while expert users found it to be of less value.

Path Following

All subjects went off the path on side-trips very often and they also found it very easy to get back on the path.

Palette use

Most subjects understood the functions of the path palette controls, although the novice users had more difficulty than the expert users. All subjects found the palette easy to use and liked using it. Two users (one novice and one expert) indicated a preference for using HyperCard's navigation facilities rather than the palette. This may be due to the increase in functionality offered by HyperCard's facilities.

Discussion

This section analyses possible reasons behind perceived trends and interesting features that were observed in the results of the study. An obvious trend is that experienced subjects had fewer navigational problems overall than novice subjects. This is to be expected and is not particularly interesting. Other trends and features are discussed below.

Feeling Lost

It is interesting that users in both groups expressed similar feelings of being lost. The path group was expected to be less lost due to the stabilising structure of the path — the nodes on the path would correspond to landmarks that the user could easily return to. In comments afterwards, several of the subjects in the path group expressed their desire for further information about the path. The main requests were that the palette provide three extra items of information: the name of the path currently open, the length of the path and the position of the current node in relation to the length. These appear to be needed to provide to the user an indication of the scope of the path and their position in it. This would enable them to better evaluate the path when they are under constraints, such as time, as was the case in this study.

The constraint of time and the lack of path scope information resulted in one experienced subject following the path very little. He still scored well on the comprehension exercise, however, and this can be attributed to his prior computing experience. As previously stated, he expressed more confidence in his own ability to find the information than he did in the path to guide him to it.

Slight modification to the palette would provide better contextual information and so decrease the feelings of being lost for the path group. This should be tested further, however.

All relevant information

It was very interesting that the path group expressed a great deal of confidence that they had seen all relevant information in the system. It was noteworthy because, for the most part, the path didn't provide all of it. This confidence has implications for the way that subjects explored. From observation, only the expert subjects were likely to effectively use HyperCard's facilities to find relevant information. Most subjects used the embedded facilities to explore, which in a large system is likely to be ineffective. Subjects in the path group had a better basis for exploration, in the fact that related material was located in proximity to nodes on the path. So exploration of the immediate surroundings was likely to be more rewarding for the path group than for the home group. This is an indication that a well-designed path can be used as a guide and is preferable to free exploration, especially for inexperienced users who often have to try and understand the 'system' rather than concentrating on the content contained within the system. The path facility can provide a simple yet effective guide to areas within the system.

Meta-information

Novices wanted more information about what they could do at any stage. This illustrates one of the reasons behind a meta-information function. The meta-information button on the path palette can provide information for the current context (i.e. the current node) that can be used for a variety of purposes. Novice users needed more information about such things as the function of the embedded facilities in a particular stack. This could be provided whenever the path runs into a new stack that has some significantly different controls. When new controls are presented an experienced user is more likely to experiment and be able to understand each function based on their experience, while a novice will be less likely to ascertain the functions correctly without some guidance. The effort that any user puts into experimentation in order to understand the controls increases their cognitive overheads which detracts their attention and processing power away from the subject matter.

The path group used the meta-information very often and the novices found it to be very useful. So this indicates that it is a useful function in addition to the actual content of the nodes.

Conclusions

The main conclusions that can be taken from this study are:

1. That inexperienced users need help more than experienced users, and the path tool can be useful in providing this.

2. That meta-information is seen as valuable by users. This includes scope information that can provide extra medium-scale contextual help.

3. That a path facility may be a useful aid for navigating complex systems, because it can provide an expert's guidance through the system as well as providing extra information such as the author's point of view.

4. A simple and consistent path control device (such as the path palette) is useful because it provides a simplified and consistent interface which is valuable for inexperienced users.

Further Investigation

This was a pilot study into the effectiveness of a path facility for navigating complex systems. As such it should be treated as a trial, with the results only perhaps a guide for further investigation. Here are a number of items that warrant further investigation after this study:

1. How effective is scope information in aiding navigation? Some users were frustrated in not knowing how long the path was, or how much information there was. This affected their navigational decisions because they were under time constraints. Providing an idea of the number of cards on the path would at least enable them to make better-informed navigational choices.

The effectiveness of providing scope information could be tested by having two groups using paths, but one using a modified version of the palette that shows the name of the path, its length, and the current position in the path. This might result in a reduced sense of being 'lost' compared with the current study.

2. How effective is meta-information? This might be tested using the paths facility with one group having meta-information available and one group not having it. Of course, the relative effectiveness of the meta-information is a result of what is contained in it and how it relates to the content contained in the path. The quality of meta-information could perhaps be evaluated as to its effect on subject's performance, both in comprehension and in subjective usability.

3. How can link markers be effectively provided within HyperCard so that 'invisible' buttons may be readily apparent as links? To test this two groups could be used. One group navigates a system that contains invisible buttons while the other group navigates the same system, but with some sort of link marker indicating the link location and link extent. To do this a mechanism that would record all mouse selection actions would be required, so that the work involved in finding links could be evaluated.

4. What effect does prior experience with the Paths facility have on performance when it is used again? Is the concept of a separate structure overlaid on top of HyperCard simple to remember? What effect does previous use have on the amount of exploration?

5. Can people use the authoring component effectively, and how does authoring a path affect comprehension? This could be examined by using two groups. One group uses a path that has been made up for them on a topic, while another must make their own path up on the same topic. Comprehension could then be tested for each group. Evaluation of the created paths compared to comprehension could also be carried out.

These are interesting questions that require further investigation to ascertain the possible use and value of a path facility to many users.

Conclusions and Future Research

Introduction

This chapter contains some conclusions about the path mechanism as an aid for navigating large hypermedia systems. Some appropriate tasks for the path facility are proposed, and a suggestion for future researchers of this area, that might enable them to better analyse their results, is suggested.

Future enhancements that would increase the effectiveness of this particular path facility are proposed. These are the provision of scope information in the path palette, the ability to open other paths from within a current path, a graphical path structure editor, the ability to provide branching paths, the provision of a path preview facility, and the possibility of accessing HyperCard cards over a network which would enable distributed hypermedia.

Then some future research that may be integrated into the idea of paths is discussed. This includes the role that intelligent agents might play in our creation and use of paths, and the use of colour and animation in providing a more effective interface to the paths facilities.

Finally, some final conclusions about the use of paths in navigating hypermedia systems are presented.

Conclusions About the Paths Facility

A path mechanism can be a useful aid in navigating hypermedia systems. The effective use of it depends on the task it is used for. Appropriate tasks would seem to be:

- A guided tour as an introduction to a subject area;
- A path reminding aid that can provide assistance in re-finding particular nodes in a system;
- A trail-blazing tool to create personal paths through a complex information space;
- A contextual reminder — the path may provide a sense of past context, by providing an interaction history that can be retraced either manually or automatically.
- A research and teaching tool, because the inherent association ability of hypermedia means that as systems become larger, the number of implicit linkages between nodes becomes greater. A paths facility can enable new trails of thought to be recorded quickly, without necessarily providing new subject matter.
- A structural filtering device. The path is overlaid on top of the system structure so it effectively filters out the information presented to the user, so the system appears much less complex than it actually is.

As a learning aid, a paths mechanism can be used in at least two distinctly different ways. One is as a guided tour through an information base giving an inexperienced student a structured path as a base on which to build. Another way would be to get a more experienced student to create their own path on a topic through an information base. They would need to provide their own annotations as meta-information to explain the path. This may give a deeper understanding of a subject than the less interactive method of just following the path.

However, assessing the actual effect of a paths mechanism as an aid to learning is complex and probably needs real-life use over a period of time for positive effects to appear. The study at Brown University showed the beneficial effects of using Intermedia in the students' analysis, synthesis, and evaluation skills [Landow, 1989]. The paths facility could be an effective addition to HyperCard and promote exploratory learning by indicating the connectivity between topics, and so promote an investigative frame of mind in students.

As a personal information filtering device and presentation tool, the paths facility can be thought of as an extension to the Personal Browser which would create a node for a card that was visited more than a threshold number of times [Monk, 1989a]. A personal paths stack might contain a number of paths that have been found useful and are saved for later reference and re-use. If so desired, these could be re-used by others, providing they had access to the same information base.

The paths mechanism is useful and could be developed and researched further. A number of enhancements that should occur are discussed in the next section. The pilot study that was carried out suggested areas for further investigation, as discussed previously. In order to examine more completely how people use a hypermedia system and a paths facility within it, better recording of the user's interaction history should take place.

Collection of Interaction History for Analysis

To provide a better analysis, each user's interaction history should be collected. This should include keystrokes, location of mouse clicks [Salomon, 1990a], cards visited, and length of visit. So a possible structure might be:

Card Identification
Interaction List
{ Keystroke / Mouse click;
location;
time }
Length of visit

This could be kept in the history list with little change to its structure. Only the recording mechanism would need to change, although this may be difficult. When such an interaction history has been recorded, analysis of the number of interactions on each card and the time taken to examine each card may better reveal the positive effects that a path facility combined with meta-information can produce.

Future Enhancements

Many future enhancements seem possible for the paths facility. One of the first is to enhance the functions available on the floating palettes to provide better contextual information, as requested by many of the subjects in the study. To give a better understanding of what path a user is on and where the current location within the path is, two additional items of information should be added to the palette — the current path name and the current location within the path. These could easily be added to two fields on the bottom of the palette so the user knows at all times what path they are following, how long the path is, and how far down the path they currently are. So the palettes might then appear something like this:

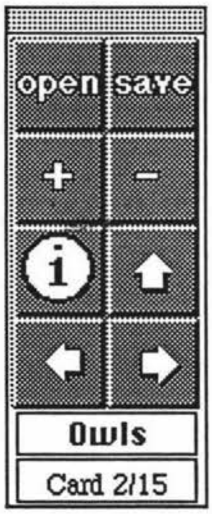


Figure 7.1 *New Author Palette, showing scope information*

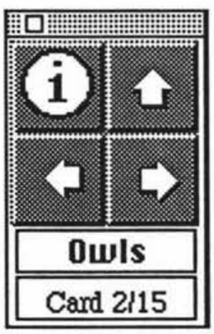


Figure 7.2 *New Student Palette, showing scope information*

The palette should also be altered to indicate what functions are available at any stage. For example, if no path is active, then the meta-information, next node and previous node buttons should not be active — they should be 'greyed out'. If there is no meta-information for a node, then it should not be selectable. And so on.

Opening Paths from within Paths

At the moment, only one path can be followed at a time. In order to open a new path, a specific path must be selected. To enhance the functionality of the paths mechanism, it might be a good idea to provide the ability to open another path from within a script rather than explicitly having to choose one. This would enable authors to seamlessly integrate the path mechanism into their stacks by opening paths from within scripts according to user actions.

For example, if a user has been making errors in some type of question & answer stack, then a guided tour might be suggested to illustrate the correct way of doing things. Similarly, a manual might have a number of guided tours incorporated within it that give examples of how to do things correctly. Of course, different guided tours may contain common material, but the sequencing of the material may alter the meaning contained in the nodes. The meta-information provided at each node may also serve to reinforce the meanings behind the links.

Structure editor

A graphical browser to display the path through the system would also be a useful addition. Such a browser could perhaps be adapted to provide a graphical path editor so that the path could be easily altered through the direct manipulation of iconic representations of the nodes. To aid this, the meta-information for each node should be stored with the path information, so re-organisation of the path does not require the simultaneous re-organisation of the meta-information structure.

Branching paths

Another facility that may be useful would be the provision of branching paths. This means that from one node there might be a number of destination nodes, and the user would have the choice of which one to follow. This has not been implemented as the present facility was originally designed to be a simple structure designed for novices, and so reducing the navigational decisions at any point was one of the aims. However, if the paths facility was to be more useful, the ability to provide multiple branches would be seen as desirable as the path-follower may tend to become more involved and this may lead to better understanding. The use of Intermedia at Brown University would tend to support this [Landow, 1989].

Path Preview

The role of path previewing is quite important. Following the HCI guideline of progressive disclosure, a path should be able to be previewed to see whether or not it is relevant to a person's needs. Of course, other tools are also helpful for getting an idea about the contents of a path — such as an introduction or summary. But a path preview is helpful in that it can be self-contained — it is not a separate entity from the path itself — it is just the manner of interaction that changes.

There are various ways one might carry out path previewing, but one of the most promising might be automatic path replay — that is, the linear progression of cards one after another after a specified period of time, just as in a slide show. This would perhaps give the user a relatively quick idea of relevancy as well as a way of assessing subjectively other values, such as presentation.

The history list could also be used to provide context. When a user gets lost or disoriented in a system it is often very helpful to show how they got to the current situation. It is also helpful to review a previous session to quickly gain context. An animation of the most recent steps could be helpful in both of these situations and could be achieved in a variety of ways. One way would be to display, in temporal order, the nodes visited and links selected on each node. Another, perhaps complementary way, would be to animate the path taken through the information space on a graphical browser, so that the nodes and links followed are successively or progressively highlighted. This might be useful in displaying, on a more global browser, the extent of the current area already visited. It would be useful on a large display on which the user might display the node contents and the changes from node to node, as well as the browser window which shows the animated path through the network.

Accessing Nodes Over a Network

An enhancement that would more easily enable paths to be shared is the ability to create paths that access nodes that are physically located elsewhere. The inclusion of relevant location information in the Path data would provide a simple means of access. Ensuring efficient access would perhaps be more difficult. Another positive aspect of paths being created over a network is that, although the information base — the collection of stacks — may not be written to at the same time by multiple users, many users can access the same node at the same time. So many personal paths can be created at the same time through a common information base, and because the meta-information is stored with the path information, annotation and modification to the stored data is controlled.

The addition of these enhancements to the paths facility for HyperCard would provide a much more powerful system. It would enable the ever-growing amount of stacks to be more effectively used as they could be easily linked to present the information in new ways — in ways that may not have been thought of at the time the information was gathered. And the paths facility also provides a simple but effective means of navigating through a complex space. Combining this with a history list that can be used for further path generation, as well as a structure editor for rapid path re-structuring, would provide a much more powerful system that would provide easier access for a wider range of users.

Future Research

Agent Oriented Systems

Kay (in Greenberger, 1990) said that computing in the '90s will not be object-oriented, rather it will be agent-oriented. Our systems may include a number of intelligent agents which communicate with each other. One agent might be a presentation agent that presents our data (perhaps called an interface agent, although presentation is better because interfacing is going to be done between all agents). Another agent might be a personal agent that keeps information about personal preferences and interests. Another agent might be a trail-blazing agent, which might contain knowledge about finding information. These three types of agent could be used in aiding navigation in Hypermedia systems. The trail-blazing agent might find information that is of interest to us, and the presentation agent could present it in a suitable way — perhaps using a path-type structure if personal preferences indicate that we are novices in the subject area.

With an intelligent agent an individual's information space can be personalised so that information of interest to each person is automatically gathered for use as needed. At the MIT Media Lab they have developed the 'personal newspaper' NewsPeek [Brand, 1987] that every day will gather interesting information from various media and present a personalised news service. Thus everyone can have their own personalised news service. What is important to a specific individual will be on the front page, for example personal mail or the cancellation of an important meeting. Less important items, such as world news, might appear later in the news. This is an example of personalised automatic selection and filtering of various media. It is also an example of personalised hypertext — it provides links to information that is important to you. Of course, it adds capabilities that should and will be included in future systems, such as automatic layout, interface elements such as the ageing of articles being indicated through colour changes.

This idea of intelligent agents can be extended from the idea of one personal agent to having a number of agents for different tasks — or perhaps having one controlling agent that can direct others. Thus we could specify that we want to be entertained and the entertainment agent could present our own personal entertainment show. Of course, this can be presented through a hypertext system — within the entertainment area a number of choices would be available and the creation of a path through that area could take place. This might include comedy, sport, music, adventure, travel, action and so on. Choosing which to follow would depend on our mood at the time, and the agent has selected items within those divisions that are interesting to us. The agent should also provide assistance to us depending on our aims. For example, several levels of interaction exist which affect the style of interaction and the agent should automatically alter the system interface according to the level chosen. Examples of these levels are [Myers, 1988]:

- Tell me — give me the facts
- Inform me — facts plus optional background and points of view
- Guide me — let me browse, but give me extra advice
- Teach me — step by step guidance
- Challenge me — make me find creative connections
- Amuse me — find interesting connections or perspectives

The way navigation takes place and the cues presented change depending on the style of interaction that is chosen, so perhaps an interaction agent that manages the interface aspects is required. The interface agent might also need to distinguish among multiple users to enable varying preferences and styles to be catered for [Laurel, 1990].

The use of software agents is already under investigation. The Object Lens project at MIT enables users to create agents which sort mail, issue reminders, and search databases [Crowston and Malone, 1988]. The utility that such aids can offer seems great, and when combined with a hypertext system the usability of such a system would increase greatly. The agents could do much of the routine tasks of filtering and selection and leave the user to examine the items that were personally relevant. Of course, the user is always able to ignore the agents' choices if they wish to pursue other options or, in keeping with our navigation metaphor, follow other paths. It certainly needs to be explored further.

Animation

Animation could be used to provide a variety of navigational cues. Baecker and Small (1990) state that

> *"Animation can:*
> - *Review what has been done*
> - *Show what can be done*
> - *Show what cannot be done*
> - *Guide a user as to what to do*
> - *Guide a user as to what not to do"*

They describe eight uses of animation of function at the interface, of which identification, transition, demonstration, explanation, feedback and history could all be used to aid navigation and orientation in hypertext systems.

Animation could be useful in identifying a link destination. For example, if the link points to a video clip or an animation itself, then a short animation used as a preview could be helpful in identifying what it is exactly and help the user in deciding whether to follow it or not.

Transitional animation is useful in hypertext systems in keeping the user aware that a change to the environment is taking place, whether the change is simply going to the next node, or whether it is a jump to another branch, or whether the user has decided to 'go back' to a previous node. Animated transitions can provide feedback as to what sort of change is taking place. These transitional animations are already widely used in the Macintosh interface with the zooming open and close functions when an application starts up and finishes. They also appear in Hypercard stacks with a wipe left or right indicating a page forward or back respectively. A zoom or a dissolve in Hypercard might indicate a jump to another non-sequential node, and an iris close visual effect might indicate a return to a previous location. All these can be useful to reinforce the action taken by the user and to orient the user during the transition from one process to another. They should be used consistently, however, so as not to confuse the user.

Animation as demonstration is related to the idea of identification. With animation the information content of various objects, such as icons, can be improved and consequently clarify their functions. For example, a hypermedia system contains different node types such as text, video, sound and graphics, and the links used to connect these different node types may be used to indicate what type of node is at the destination. So, as in the identification animation, a link might be represented by an icon that demonstrates, through animation, what is at the destination node. Another use for animated demonstrations would be in combination with a system like NoteCards where a user may have a number of open windows on a card, similar to the TableTop card previously described. An animation could highlight each card in order to demonstrate the order in which they are intended to be read.

Feedback is essential to any interface and animated feedback is very effective in signifying the percentage of process completion [Myers, 1985]. Obviously transitional animation is an example of animated feedback, but there are other examples applicable to navigation and orientation also. A graphical browser that is simultaneously displayed with the nodes and links may give an indication, through animation, of which link is being followed. It may also indicate how much longer it is going to take through progressive highlighting of the link on the browser. This might be applicable in a system that accesses information around the world so that the user can see how close the system is to displaying the selected node.

There are problems with providing animation of course. To be effective, animation requires careful planning and design. It can also require significant processing power, so if the animation degrades the response time of the system, the users will revert to prior, perhaps less effective, methods of information presentation. There are many possibilities to be investigated in the uses of animation at the interface as an aid to navigating hypertext systems, and these methods need greater research.

Colour

Colour can be an effective tool in adding extra information to the interface, and this can be helpful in assisting navigation if used in a consistent and appropriate manner. There are a number of problems with using colour however, and the hypertext designer should be aware of these problems. Several problems are to do with our colour perception. Colours interact with those around them so that the appearance of a colour can change depending on those around it. External conditions such as ambient lighting affect our colour perception.

Computer monitors also vary in their colour calibration so there is no guarantee that a particular colour combination will look the same on two different monitors. Another problem is the individual differences present in the population due to culture, age, occupation, and gender differences — a higher proportion of men have colour deficient vision than women [Salomon, 1990].

In spite of these problems colour can be used to good effect in aiding navigation. It can be used to identify landmarks and to indicate the passage of time, for example. These are two obvious uses for colour that can provide extra information without taking up extra screen space.

Distinguishing landmarks through the appropriate application of colour could be used in many areas and levels of navigation. Landmark nodes could be distinguished by a different background colour. This different colour could also be carried over on to the browser so that landmark nodes are quickly identified through their different colour on the map.

The passage of time could be indicated by altering the colour of a node's representation on the map when it has not been visited for a certain amount of time. A node might fade from yellow to dark brown as it gets 'older', for example. Another example might be to use a certain colour to indicate the nodes that must be visited, as when a teacher supplies a path for a student to follow. The student is free to explore all of the hypertext, but the nodes that have to be visited are highlighted through the use of a distinctive colour.

Colour has been used effectively to indicate both the type and state of nodes in the graphical browser of gIBIS. The type mappings for the most commonly used nodes seemed to be quickly learned so that type identification became a rapid, reflex activity [Begeman & Conklin, 1988]. They say that the use of colour to indicate the node types and links has been a clear success, but that this may be in part due to the limited number of node and link types available.

As Salomon (1990) indicates, some testing is required to discover suitable effective combinations as first ideas are sometimes found to be ineffective. Colour can be used in addition to other attributes such as shape, texture, pattern and location to aid identification when some perception problems might be anticipated. It is a promising area of investigation that needs intelligent guidelines for use, as initial use can result in overuse in a similar manner to the 'fontitis' phenomenon that occurred when people were introduced to the availability of multiple fonts.

Applied to the paths facility, perhaps a colour could be overlaid on top of nodes that were on the path to differentiate them from the surrounding nodes. This would give an immediate indication to the user when they were on or off the path. This could be extended on to the path browser so path nodes would be easily distinguished again. Nodes that had already been visited might change to a different colour to indicate the historical path.

These three areas — agents, animation, and colour — are very powerful and may affect our interactions greatly and positively. They need to be explored further as the importance of effective navigational facilities is increasing because our information systems and hypermedia systems are becoming larger and global in nature.

Overall Summary

When different levels of navigation, the growing complexity of hypermedia systems, and the increasing amount of information available are taken into consideration, a paths facility will be found to be an important navigational aid because it can:

- filter the information presented;
- guide users through unknown areas;
- provide advice, narrative point of view, and contextual information through the use of meta-information;
- provide a simple and consistent interface for novice users; and
- provide a means of saving personal information paths through a complex network for later re-use.

The paths facility in this thesis may be regarded as an attempt at exploring some of the issues involved in navigating large systems and bringing further questions to light. Further research should be carried out in this area as mentioned previously. To be most effective, a paths mechanism should be integrated into a hypermedia system so that it can be used in conjunction with other facilities such as a graphical browser/editor, contextual filters such as webs, existing history facilities, and possibly agents that can filter information in order to create new paths. This would result in more powerful and effective tools for a wider range of users, and the benefits of these systems would be better realised.

Bibliography

ALLINSON, L. & N. HAMMOND (1988): A Learning Support Environment: 'The Hitch-Hikers Guide', *Hypertext Aberdeen 1988.*

BAECKER, R. & I. SMALL (1990): Animation at the Interface, *The Art of Human-Computer Interface Design,* Brenda Laurel (ed.), Addison Wesley, Reading Mass.

BAIRD, P. (1988): Glasgow Online: Database Development using Apple's Hypercard, *Hypertext Aberdeen 1988.*

BEGEMAN, M.L. & J. CONKLIN (1988): The Right Tool for the Job, *BYTE,* October, 1988, pp. 255 - 266.

BERNERS-LEE, T. (1991): WorldWideWeb Summary, Internet Posting, 6 August 1991, Alt.Hypertext newsgroup.

BRAND, S. (1987): *The Media Lab: Inventing the Future at MIT,* Viking Penguin Inc., New York.

BRUNER, J.S. (1968): *On Knowing: Essays for the Left Hand,* Atheneum, New York.

BUSH, V. (1945): As We May Think, *Atlantic Monthly,* July 1945, pp. 101 - 108.

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

CROWSTON, K. & T.W. MALONE (1988): Intelligent Software Agents, *BYTE,* 13, 13, pp. 267 - 274.

DAVIES, G., MAURER, H., PREECE, J. (1991): Presentation Metaphors for Very Large Hypermedia Systems, *Journal of Microcomputer Applications*, 14, pp. 105 - 116.

DUNCAN, E.B. (1989): Structuring Knowledge Bases for Designers of Learning Materials, *Hypermedia*, 1, 1, pp. 20 - 33.

EDWARDS, D.M. & L. HARDMAN (1989): 'Lost in Hyperspace': Cognitive Mapping and Navigation in a Hypertext Environment, in *Hypertext: theory into practice*, Ray McAleese (Ed.), Intellect, Oxford, pp. 105 - 125.

ELLIS, D. (1990): *New Horizons in Information Retrieval*, Library Association Publishing Ltd, London.

ERICKSON, T.D. (1990): Working with Interface Metaphors, *The Art of Human-Computer Interface Design*, Brenda Laurel (ed.), Addison Wesley, Reading Mass.

EVENSON, S. & RHEINFRANK, J. (1989): Towards a Design Language for Representing Hypermedia Cues, *Hypertext '89 Proceedings*, ACM, New York, pp. 83 - 92.

FAIRCHILD, K., MEREDITH, G. & WEXELBLAT, A. (1989): The Tourist Artificial Reality, *CHI '89 Proceedings*, pp. 299 - 304.

FIDERIO, J. (1988): A Grand Vision, *BYTE*, October, 1988, pp. 237 - 244.

FLORIN, F. (1990): Information Landscapes, *Learning with Interactive Multimedia: Developing and Using Multimedia Tools in Education*, Sueann Ambron & Kristina Hooper (eds.), Microsoft Press, pp. 28 - 49.

FRISSE, M.E. (1988): Searching for Information in a Hypertext Medical Handbook, *Commun. ACM*, 31, pp. 880 - 886.

FURNAS, G.W. (1986): Generalized Fisheye Views, *Proceedings of CHI '86 Human Factors in Computing Systems*, ACM, New York, pp. 16 - 23.

GREENBERGER, M. (1990): *Technologies for the 21st Century: On Multimedia*, The Voyager Company, Santa Monica, California.

HAAN, B.J., P. KAHN, V.A. RILEY, J.H. COOMBS & N.K. MEYROWITZ (1992): IRIS Hypermedia Services, *Commun. ACM*, 35, 1, pp. 36 - 51.

HALASZ, F.G. (1987): Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems, *Hypertext '87 Papers*, pp. 345 - 365.

HALASZ, F.G. (1988): Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems, *Commun. ACM*, 31, pp. 836 - 852.

HAMMOND, N. and ALLINSON, L. (1987): The Travel Metaphor as Design Principle and Training Aid for Navigating Complex Systems, in *People and Computers III*, D. Diaper & R. Winder (eds.), Cambridge University Press, pp. 75 - 90.

HAMMOND, N. and ALLINSON, L. (1988): Travels around a Learning Support Environment: Rambling, Orienteering or Touring?, *CHI '85 Proceedings*, pp. 269 - 273.

HAMMOND, N. and ALLINSON, L. (1989): Extending Hypertext for Learning: An Investigation of Access and Guidance Tools, in *People and Computers V*, A. Sutcliffe & L. Macaulay (eds.), Cambridge University Press, pp. 293 - 304.

HAMMOND, N., MAYES, T. & BARDEN, R. (1989): Hypertext for Learning: Problems and Prospects, *Hypertext II Proceedings*, York.

HARDMAN, L., SHARRATT, B. (1989): User-Centred Hypertext Design: the application of HCI design principles and guidelines, *Hypertext II Proceedings*, York.

KAHLE, B. (1989): Wide Area Information Servers Concepts, *Thinking Machines Corporation technical report TMC-202.*

KAHN, P. (1989): Webs, Trees, and Stacks: How Hypermedia System Design Effect Hypermedia Content, *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, G. Salvendy & M.J. Smith (eds.), Elsevier Science Publishers B.V., Amsterdam.

KAHN, P., J. LAUNHARDT, K. LENK & R. PETERS (1990): Design of Hypermedia Publications: Issues and Solutions, *Proceedings of the International Conference on Electronic Publishing, Document Manipulation, and Typography*, Richard Furuta (Ed.), Cambridge University Press, pp. 107 - 124.

LANDOW, G. P. (1989): Hypertext in Literary Education, Criticism, and Scholarship, *Computers and the Humanities*, 23, pp. 173 - 198.

LANSDALE, M.W., D.R. YOUNG & C.A. BASS (1989): MEMOIRS: A Personal Multimedia Information System, *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, G. Salvendy & M.J. Smith (eds.), Elsevier Science Publishers B.V., Amsterdam.

LAUREL, B. (1990): Interface Agents: Metaphors with Character, *The Art of Human-Computer Interface Design*, Brenda Laurel (ed.), Addison Wesley, Reading Mass.

LAUREL, B., T. OREN & A. DON (1990): Issues in Multimedia Interface Design: Media Integration and Interface Agents, *CHI '90 Proceedings*, pp. 133 - 139.

LINDE, L. (1989): An Information Seeker's Mental Model of the Content and Search Principles in a Database, *Man-Computer Interaction Research: MacInter II*, F. Klix, N. A. Streitz, Y. Waern & H. Wandke (eds.), Elsevier Science Publishers, Amsterdam, pp. 149 - 159.

LINDERHOLM, O. (1992): Mind Melding, *BYTE*, 16, 11, pp. 41 - 46.

LIVINGSTON, K. (1991): personal communication, e-mail, IRIS, Brown University.

McKNIGHT, C., RICHARDSON, J. & DILLON, A. (1989): The Authoring of HyperText Documents, in *Hypertext: theory into practice*, Ray McAleese (Ed.), Intellect, Oxford, pp. 138 - 147.

MARSHALL, C.C., IRISH, P.M. (1989): Guided Tours and On-Line Presentations: How Authors Make Existing Hypertext Intelligible to Readers, *Hypertext '89 Proceedings*, ACM , New York, pp. 15 - 26.

MONK, A. F. (1989): Getting to Known Locations in a Hypertext, *Hypertext II Proceedings*, York.

MONK, A. F. (1989a): The Personal Browser: a tool for directed navigation in hypertext systems, *Interacting with Computers*, 1, 2, pp. 190 - 196.

MYERS, B.A. (1985): The Importance of Percent-Done Progress Indicators for Computer-Human Interfaces, *CHI '85 Proceedings*, pp. 11 - 17.

MYERS, B.A. (Ed.) (1988): *Speculations on The Personal Computer of the Year 2000*, Carnegie Mellon University Computer Science Department Report CMU-CS-88-115.

NEGROPONTE, N. (1990): Hospital Corners, *The Art of Human-Computer Interface Design*, Brenda Laurel (ed.), Addison Wesley, Reading Mass., pp. 347 - 353.

NELSON, T.H. (1988): Managing Immense Storage, *BYTE*, January 1988, pp. 225 - 238.

NIELSEN, J. (1987): Trip Report: Hypertext '87, *ACM SIGCHI Bulletin*, 19, 4, pp. 27 - 35.

NIELSEN, J. (1990): *HyperText and HyperMedia*, Academic Press Inc., San Diego.

NIELSEN, J. (1990a): The Art of Navigating Through Hypertext, *Commun. ACM*, 33, pp. 296 - 310.

OREN, T. (1987): The Architecture of Static Hypertexts, *Hypertext '87 Papers*, pp. 291 - 306.

OREN, T. (1990): Cognitive Load in Hypermedia: Designing for the Exploratory Learner, *Learning with Interactive Multimedia: Developing and Using Multimedia Tools in Education*, Sueann Ambron & Kristina Hooper (eds.), Microsoft Press, pp. 125 - 136.

OREN, T., SALOMON, G., KREITMAN, K., DON, A. (1990): Guides: Characterizing the Interface, *The Art of Human-Computer Interface Design*, Brenda Laurel (ed.), Addison Wesley, Reading Mass.

PALANIAPPAN, M., N. YANKELOVICH & M. SAWTELLE (1990): Linking Active Anchors: A stage in the evolution of Hypermedia, *Hypermedia* 2, 1, 1990, pp. 47 - 66.

PETERSEN, A.M. (1989): A Library System for Information Retrieval Based on a Cognitive Task Analysis and Supported by our Icon-based Interface, *Proceedings of the 12th International SIGIR Conference*, Cambridge, pp. 40 - 47.

REICHMAN, R. (1986): Communication Paradigms for a Window System, *User Centred System Design: New Perspectives on Human-Computer Interaction*, D.A. Norman & S.W. Draper (Eds.), Lawrence Erlbaum Associates, Inc., New Jersey, pp. 285 - 313.

REINKING, D. (1992): Differences Between Electronic and Printed Texts: An Agenda for Research, *Journal of Educational Multimedia and Hypermedia*, 1, 1, pp. 11 - 24.

SALOMON, G. B. (1990): New Uses for Color, *The Art of Human-Computer Interface Design*, Brenda Laurel (ed.), Addison Wesley, Reading Mass.

SALOMON, G. B. (1990a): Designing Casual-Use Hypertext: The CHI '89 InfoBooth, *CHI '90 Proceedings*, pp. 451 - 458.

SALTON, G. (1989): *Automatic Text Processing - The Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley Publishing Co., Reading Massachusetts.

SIMPSON, A., MCKNIGHT, C. (1989): Navigation in Hypertext: Structural Cues and Mental Maps, *Hypertext II Proceedings*, York.

STRONG, G.W. & K.E. O'NEILL STRONG (1991): Visual guidance for information navigation: a computer-human interface design principle derived from cognitive neuroscience, *Interacting with Computers*, 3, 2, pp. 217 - 231.

TRIGG, R. (1988): Guided Tours and Tabletops: Tools for Communicating in a Hypertext Environment, *ACM Transactions on Office Information Systems*, 5, pp. 398 - 414.

UTTING, K. (1989): *Some Thoughts on the Intermedia Web View*, Unpublished Paper.

UTTING, K. & N. YANKELOVICH (1989): Context and Orientation in Hypermedia Networks, *ACM Transactions on Information Systems*, 7, 1, pp. 58 - 84.

VAN DAM, A. (1988): Hypertext '87 KeyNote Address, *Commun. ACM*, 31, pp. 887 - 895.

WALKER, J. (1987): Document Examiner: Delivery Interface for Hypertext Documents, *Hypertext '87 Papers*, pp. 307 - 323.

YANKELOVICH, N., B. J. HAAN, N. K. MEYROWITZ & S. M. DRUCKER (1988): Intermedia: The Concept and the Construction of a Seamless Information Environment, *IEEE Computer*, January, 1988, pp. 81 - 96.

ZELLWEGER, P.T., (1989): Scripted Documents: A Hypermedia Path Mechanism, *Hypertext '89 Proceedings*, The Association for Computing Machinery, New York., pp. 1 - 14.

Appendices

Demographic Questionnaire

1. Gender M F

2. Age <20 20 - 30 30 - 40 40 +

3. What kinds of computer have you previously used?

Macintosh
IBM PC (or clone)
Workstation (e.g. Sun)
Other (specify)

4. Have you ever used a mouse and a graphical user interface (e.g. Windows)? Y N

5. Have you ever used Hypercard or a Hypercard-like program (e.g. Toolbook, Plus)?
Y N

6. How experienced are you with computers?	Not at all	Some	Average	Medium	Very
7. How experienced are you with Macintosh?	Not at all	Some	Average	Medium	Very
8. Do you know much about Owls?	Not at all	Some	Average	Medium	Very
9. Do you know much about bird physiology?	Not at all	Some	Average	Medium	Very

10. What is your level of education?

High School
Polytechnic
Degree (specify)

Instructions — Path Group

Introduction

Welcome to this small pilot study. I am conducting some research about the usability of hypertext and finding information within hypertext systems. I appreciate your assistance in taking the time to carry out the task and to answer the questionnaires. Thank you.

Before you begin, please fill out the demographic questionnaire if you have not already done so.

If you are not familiar with Hypercard and using the Hypercard facilities to find information, please use the stack Introduction to Hypercard Navigation to familiarise yourself. This will give you an idea how Hypercard is structured and various ways you can find information of interest to you within a Hypercard system.

Please see your reference sheet *Guided Paths Introduction* for an description of the guided paths facility that you will be using.

Your Task

You must use the guided paths facility to find out all you can about Owls and also about general characteristics of birds. These include general surface characteristics, feather anatomy and flight.

You should follow the Owls guided path. This will present you with a floating palette that enables you to go back and forth along a guided path. This path will incorporate information from various stacks - you do not need to be concerned where the information is from. Concentrate on finding out as much as you can on the topics. You are free to explore off the path using Hypercard facilities as you wish.

To understand the palette controls, please feel free to read the Palette reference sheet now, and you may refer to it at any stage during your exploration.

Hints

A couple of hints or reminders may be appropriate. Remember that the information button will provide extra help about what you should be looking for, what to do next and so on. If you explore using the facilities of Hypercard, remember that the *Jump back* button will take you back to the guided path.

Now

Please start up your Owls guided path by:

1. Open the *Guided Paths* stack by double clicking on it.
2. Click on the opening card.
3. Select the **Owls** guided path and select **Ok**.

Then you will see the first card in the path and the Paths floating palette which is your control centre.

You have 30 minutes to complete your exploration. At the end of this time you will have a couple of short questionnaires to complete.

Instructions — Home Group

Introduction

Welcome to this small pilot study. I am conducting some research about the usability of hypertext and finding information within hypertext systems. I appreciate your assistance in taking the time to carry out the task and to answer the questionnaires. Thank you.

Before you begin, please fill out the demographic questionnaire if you have not already done so.

If you are not familiar with Hypercard and using the Hypercard facilities to find information, please use the stack *Introduction to Hypercard Navigation* to familiarise yourself. This will give you an idea how Hypercard is structured and various ways you can find information of interest to you within a Hypercard system.

Your Task

This system contains a small number of Hypercard stacks. You must explore them using the available Hypercard facilities and find out all you can about Owls and also about general characteristics of birds. These include general surface characteristics, feather anatomy and flight.

You will have available a Home card from which all the stacks that you require are accessible. Not all the stacks necessarily contain information relevant to your task, but you are free to explore any of these stacks as you wish. Please use the Home Reference sheet anytime during your exploration. This contains information that may assist you.

Hints

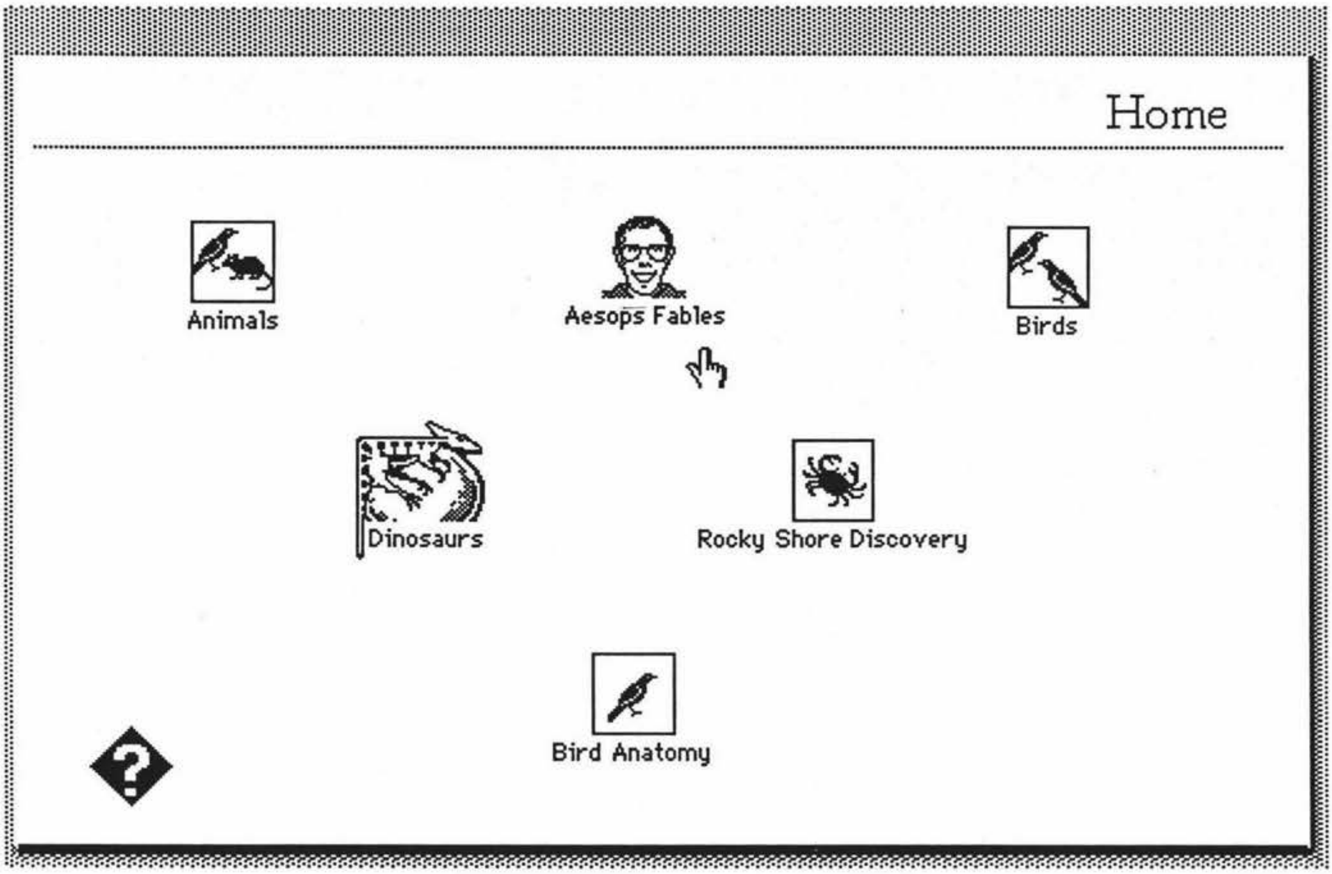
A couple of hints or reminders may be appropriate. Remember that the **Go** menu in Hypercard assists your navigation task, and the **Find** item in the **Go** menu can assist you in finding textual information. Remember also that there are often invisible buttons over pictures and that clicking on them may provide more information.

Now

You have 30 minutes to complete your exploration. At the end of this time you will have a couple of short questionnaires to complete. Please start up your Home card now by double clicking on the Home stack icon.

Home Reference Guide

This is the card that you will see on first opening the stack. From here you can access any of the other stacks by clicking once on any of the icons.



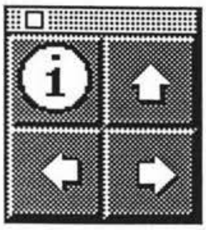
Click once on the question mark icon to get information about what to do.

In most of the stacks there will be a small icon representing a building, similar to this. This is called the home icon. Clicking on it will bring you home, which is this card here. So it can be used as a quick way of getting back to this 'control centre' from which you can explore the rest of this system.

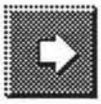
Hypercard itself provides other controls which you should use to locate information. They are all located under the **Go** menu. With these controls, you can go from one card to the next (or previous) and go to the first or last card.

To find text information quickly within a particular stack you should use the **Find** command under the **Go** menu. Select the **Find** command and you will see the message box appear with the words Find "". The cursor will appear between the quote marks. Now you should type in the information that you wish to find and then press *return*. The text that you wish to find will be highlighted by a square box around it. To find other occurrences of the same text, just press *return* again.

Path Palette Reference



This palette provides four functions for you. Each function is accessed by clicking once on one of the four icons. You will see some action occur such as the turning of a page on to a new screen, or a small information window appearing.



The right pointing arrow takes you to the next card in the path. When you reach the end of the path you will not be able to go any further. You can retrace your steps with the left arrow or open the path again to start afresh.

The left pointing arrow takes you to the previous card in the path.

This enables you to go back to any previous card in the path i.e. to retrace your steps.

The up arrow takes you to the current card in the path.

This is useful for when you have used the controls within a stack to explore a little off the path. If you want to quickly return to the path at the point where you left it, clicking on this icon will return you there.

The information icon provides meta-information. That is, it provides information about what you're supposed to do, why you have been taken to a particular card, or a hint about what to do on a particular card.

When you click on the information icon, a small window will appear with some informative text inside it. You can move this window anywhere on the screen so you can put it out of the way. You can also resize the window. To close the information window, click on the close box in the top left corner of it.

Guided Paths Explanation

Guided paths are a new structure overlaid on top of the normal Hypercard structures. A path is a number of cards connected up in linear fashion and allows you to access information from various different stacks very easily. A floating tools palette provides you with access to the path - you click on the various tools to use the path.

Figure 1 shows the guided paths idea.

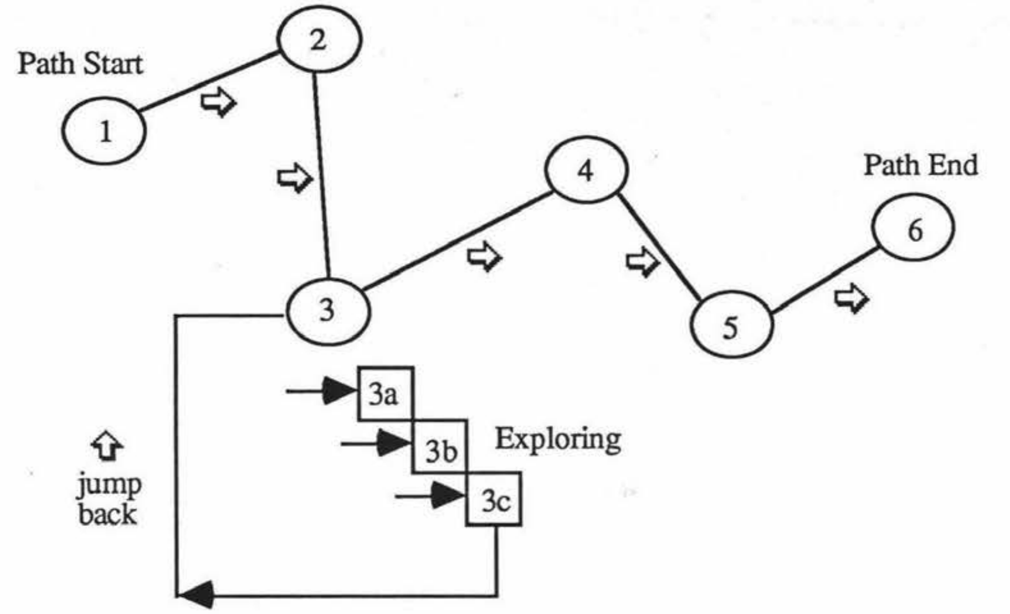


Fig. 1. Guided Path, Exploration and Jumping back

Notes: The Circles represent cards on the path and each of these can be located in any stack. The squares represent cards that are not on the path.

1. In Fig. 1 we see that Card 1 is the start of the path. Using the palette's next card button we follow the path from card 1 to card 2 to card 3.

2. At card 3 we find something of interest and we decide to explore this area a little further. So then we use the inbuilt navigation facilities of that particular stack to go forward three cards - 3a, 3b, 3c - that are not on the path.

3. After taking that little side-trip we decide to return to the card that we left the path from. We do this by using the jump-back button (the arrow pointing up) on the palette. This returns us to card 3.

4. Then we continue following the path to card 4, card 5 and card 6.

Post-Test Questions - Usability

Please circle the number on the five point scale for each of the following questions.
e.g. For question one, if you felt that you were *never* lost, you would circle the number 5.

1. Did you feel lost at any stage? very lost 1 2 3 4 5 not at all lost

2. Did you feel that you saw all there was in the system about Owls?
No 1 2 3 4 5 Definitely

3. Were you confused at any stage about what to do next?
Very 1 2 3 4 5 Not at all

4. Did you understand the navigational facilities on each card?
Never 1 2 3 4 5 Always

5. Did you ever want more information about what you could do on a card?
Never 1 2 3 4 5 Often

6. Were any of the controls confusing? Never 1 2 3 4 5 Often

7. How would you prefer to explore the system? Using:
   - a. a floating palette of tools
   - b. a menu
   - c. hypercard facilities such as buttons
   - d. typed commands using a keyboard
   - e. verbal commands
   - f. other (specify)

8. What other problems did you have? (specify)

Post-Test Questions — Path Usability

Please circle the number on the five point scale for each of the following questions.

1. How much of the path did you follow? Not much 1 2 3 4 5 All

2. Did you use the information button at any time? Never 1 2 3 4 5 Often

3. If you used the information button, how useful was it?
No use 1 2 3 4 5 Very useful

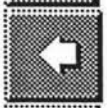
4. Did you go off the path at any time? Never 1 2 3 4 5 Often

5. If you went off the path, how easy was it to get back on the path?
Difficult 1 2 3 4 5 Easy

6. Did you understand the controls on the palette?
Not at all 1 2 3 4 5 All of them

7. Would you prefer to use guided paths or Hypercard's facilities (e.g. buttons, the find function, the Go menu, etc.) to find information?
Palette Hypercard's facilities

8. Please circle the best description of the function of the following buttons from the palette:

	go to next page	go right	go to next in path	go to last in path
	go to previous page	go left	go to prev. in path	go to first in path
	go up one level	go to first in path	go to last in path	go back to path
	help	information	statistics	make notes

9. Was the palette easy to use? Difficult 1 2 3 4 5 Easy

10. Did you like using the palette? Hate 1 2 3 4 5 Loved

11. Did you have any other problems with the palette? (specify)

Comprehension

1. The Snowy Owl is mainly active during what time of day?
   - 1.1. dawn & dusk
   - 1.2. daylight hours
   - 1.3. nighttime
   - 1.4. any time

2. The Screech Owl is mainly active during what time of the day?
   - 2.1. dawn & dusk
   - 2.2. daylight hours
   - 2.3. nighttime
   - 2.4. any time

3. The eyes take up what proportion of the Owl's head? About:
   - 3.1. 1/4
   - 3.2. 1/3
   - 3.3. 1/2
   - 3.4. 2/3
   - 3.5. 3/4

4. What effect does the phase of the moon have on Owl's behaviour?
   - 4.1. Breeding patterns
   - 4.2. hunting
   - 4.3. more active during fuller moon
   - 4.4. all of the above
   - 4.5. no effect

5. The ear tufts of Owls affect what?
   - 5.1. vision
   - 5.2. hearing
   - 5.3. radar
   - 5.4. none of the above

6. The Rachis is in what part of the owl:
   - 6.1. the feathers
   - 6.2. the wing
   - 6.3. the eye
   - 6.4. the foot
   - 6.5. the head

7. A bird's flight is explained by what principle:
   - 7.1. Newton's Laws
   - 7.2. Bernouilli's Principle
   - 7.3. Fitt's Law
   - 7.4 Fourier's Law

8. What is the main reason that birds are able to fly?
   - 8.1. Strength of the wings
   - 8.2. Shape of the wings
   - 8.3. Strength of the feathers
   - 8.4. Shape of the feathers
   - 8.5. Air currents

9. What purpose do down feathers play?
   - 9.1. To keep out water
   - 9.2. To assist flight
   - 9.3. To protect the young
   - 9.4. To trap air

10. The Wing Chord is:
   - 10.1. The span of the wings
   - 10.2. the wings supporting structures
   - 10.3. the width of the wings
   - 10.4. the ratio of wing span to wing width

11. What structure is unique to birds' eyes?
    - 11.1. the Pecten
    - 11.2. the Sclerotic ring
    - 11.3. the Sclera
    - 11.4. the Vitreous Humor

12. Where are the Remiges?
    - 12.1. on the head
    - 12.2. in the eye
    - 12.3. on the beak
    - 12.4. in the wings
    - 12.5. in the feet

13. According to Aesop's Fables, Owls are regarded as:
    - 13.1. Ignorant
    - 13.2. Annoying
    - 13.3. Wise
    - 13.4. Inquisitive
    - 13.5. Clever

The Paths Facility – How to Use it

The paths facility provides different functions for two groups of users which are termed Authors and Students, based on the dichotomy between a path-maker (author) and a path-follower (student). Of course, there will not always be this separation but in a learning situation there often is. So in the next sections the functions available to each group of users will be described.

Authors

Authoring requires a number of main functions. The primary ones are the ability to create and edit a path, and the ability to add meta-information. These functions are provided on the author's palette and so are just a mouse-click away.

To get into Authoring mode a user should type *Author* into the HyperCard message box and press return. This will result in the Author palette appearing. From there all the Author functions that are described next can be accessed.

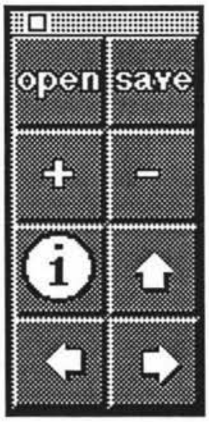


Figure 1 - *Author Palette*

Creating a New Path

In order to create a new path, the author clicks on the open button on the author's palette. This will bring up the dialog in figure 2. To create a new path, the author should click on the New button. After doing this, figure 3 will appear. Here the author enters the name of the new path and then clicks on the Ok button. If the author decides not to create a new path now, then they can stop the path being created by clicking on the Cancel button.

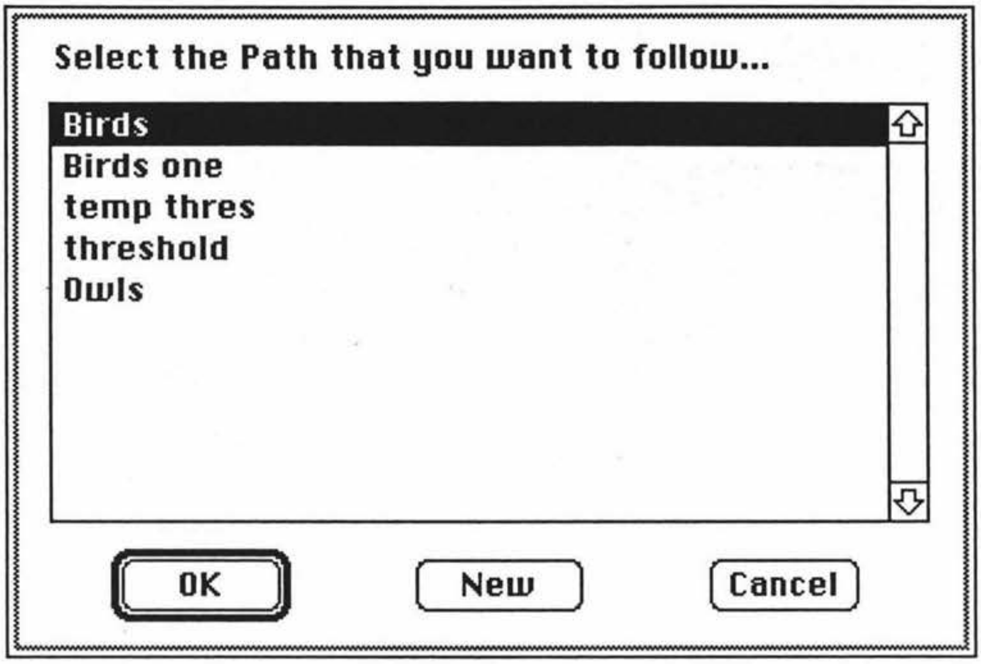


Figure 2 - *Open Path Dialog*

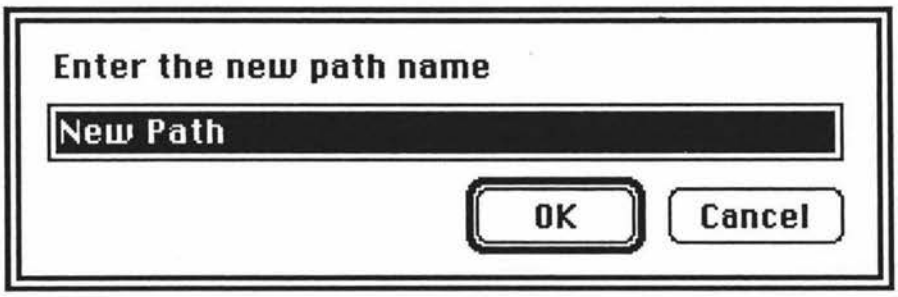


Figure 3 - *Name Path Dialog*

When a new path is created, new structures are created to provide storage for this path and its associated information. These structures are:

- the Path
- the Meta-information
- the History

At the moment, these structures are provided as fields on one card of a Hypercard stack. One card in the paths stack serves each path that is created. When a new path is created, a new card is created with fields called "*Path*", "*Metainformation*", and "*History*". The name of the path that the user entered is set as the name of the card.

Opening an Existing Path

To open an existing path, the author should click on the open button on the palette. This will bring up figure 2 which shows a list of the existing paths. To open one of these, the author should select one of the paths, either by clicking on it with the mouse or by using the cursor control keys to move up and down the list. Then the author should click on the Ok button to open the selected path.

When a path is opened, the appropriate structures will be loaded into memory from the card corresponding to the selected path. The author will then be taken to the first node in the path.

Adding to the Path

To add to a path the user should click on the + button on the palette. This will add the current node to the path. The current card's details are inserted after the current node in the path structure and the current node is set to the newly added node. Space is also provided in the meta-information structure for the author to add meta-information about this node.

Deleting from the Path

To delete the current node from the path a user should click on the – button on the palette. Before deletion, the user will be asked to confirm their intention as in figure 4. If Cancel is selected, then they will be returned to where they were. If Ok is selected then the current node will be deleted from the path as well as the meta-information associated with this node. The current node will then be set to the next node in the path. If the node at the end of the path is deleted then the current node will be the one prior to the node that was deleted, and it will be the new end of the path.

A special case exists if the embedded navigation facilities in the Hypercard stacks are used to navigate to a node that is not on the path and is therefore not the current node. If the – button is then selected the current node will be returned to and the user will be asked to confirm that this is the node they want to delete, as in figure 4.

Figure 4 - *Confirm node deletion dialog*

Saving the Path

As all operations on a path occur in memory and do not occur on the actual path fields, then a save operation must be carried out to explicitly save the path.

To save the current path the user should click on the save button on the palette. This will bring up figure 5. Three options are available: Cancel, Overwrite, or New.

Figure 5 - *Save Path dialog*

Selecting Cancel will not save the path and will leave the user at the same position that they were before selecting the Save button. No action will be performed.

Selecting Overwrite will overwrite the existing path with the path that is currently stored in memory — that is, it will save any changes that have been made to the current path. For example, if an existing path was originally opened and some nodes added and deleted from it, then selecting Overwrite will save the changes made to that path, with the old path no longer being accessible. If a new path was originally created, then selecting Overwrite will save the path that was created into the new structures that were created.

Selecting New will create a new path into which the current structures will be saved. The user will be asked to name the new path as in figure 3. After naming the new path, the current path will be saved to this new path. This enables a user to open up an existing path, make changes to it, and then to save the altered path to a new path. This means that the original path can be saved as well as the new path.

Whenever a path is saved, not only the path information is saved, but also the meta-information associated with the path is saved.

Meta-information

When an author clicks on the Meta-information button on the palette, two things may happen. If there is already information for the current node then that information will be displayed. If there is no information for the current node, then meta-information can be added for the current node. The small meta-information window will appear and the user may then insert the cursor and start typing in their information. The information window is a small external window that can be moved, re-sized, scrolled and closed.

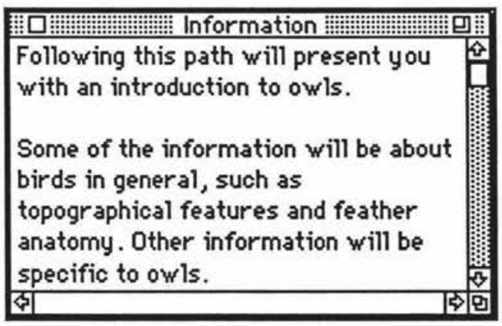
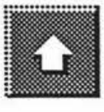


Figure 6 - *Meta-information Window*

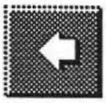
When the author goes to another node the window will be updated to show the information for the new node. If no meta-information exists for the new node, then the window will be cleared ready to accept new meta-information.

Jumping to the Current Node



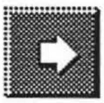
When the author has been exploring the system and wishes to return to the current node they can choose the "Jump Back" button on the palette. This will return them to the point in the path that they left it from.

Going to the Previous Node



To go to the previous node in the path, the author selects the "Previous Node" button on the palette. If the author is at the first node in the path, then a message will be displayed and they will remain where they are. Otherwise, they will go to the previous node in the path and it will become the current node.

Going to the Next Node



To go to the next node in the path, the author selects the "Next Node" button on the palette. If the author is at the last node in the path, then a message will be displayed and they will remain where they are. Otherwise, they will go to the next node in the path and it will become the current node.

Students.

Students have a limited subset of the commands available to Authors. This is to reduce their attentional demands so that they only need to concentrate on the path contents rather than on manipulation of the path contents. All their processing power should be directed towards understanding the path contents and the meta-information.

The commands available to Students are: Meta-information, Jump back, Previous Node, and Next Node. There are only minor differences in operation and these are explained below.

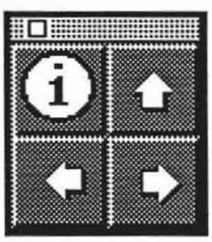


Figure 7 - *Path palette for students*

Meta-information

A student does not have the ability to add or edit the meta-information. They may display, move, re-size and close it only.

The other functions work as previously described in the Author's section.

Guided Paths Script

```
on openCard
  -- this is so that every card that is visited gets added to the
  -- History list. But if a card or other stack already has an
  -- openCard handler then it won't be added.
  global myHistory
  AddHistory (the short id of this cd), (the short name of this stack)
end openCard

on openStack
  global myPathName, myPalette
  put "Test Paths" into myPathName
  if myPalette is empty then student
end openStack

on author
  global myPalette
  if there is a window myPalette then close window myPalette
  put "Author" into myPalette
  palette myPalette
end author

on student
  global myPalette
  if there is a window myPalette then close window myPalette
  put "student" into myPalette
end student

on openPath
  global myPath, myMetaInfo, myHistory, currNodeNum, startTicks, myPathName
  global myInfoWindow, myInfoRect, myDelim, infoEditStatus, myPalette, openPathStatus
  --load contents of this stack into global variables
```

```
  lock screen
  push cd
  pop cd into myCurrCd
  set lockMessages to true
  put "Information" into myInfoWindow
  put "30,30,200,120" into myInfoRect
  put "endNode" into myDelim
  put "FALSE" into infoEditStatus
  -- put "student" into myPalette

  start using stack "Test Paths"  -- makes the scripts accessable from other stacks
  getPath
  palette myPalette

  -- -- now display
  if openPathStatus = "cancel" then
    go to myCurrCd
  else
    put 0 into currNodeNum
    put the ticks into StartTicks

    if myPath is not empty then
      nextPath  -- this activates the next node in path handler
    end if  -- which is located in the Home stack

    -- go to cd id (word 1 of line 1 of myPath)
    unlock screen

    -- set the information palette window position just
    -- to the right of the "Guided Paths" palette if it's there
    if there is a window myPalette then
      put (the right of window myPalette + 5) into myRight
      put (item 1 of myInfoRect + myRight) into item 1 of myInfoRect
      put (item 3 of myInfoRect + myRight) into item 3 of myInfoRect
    end if
  end if
end openPath
```

```
on getPath
  -- select a path to follow from the available paths stored in the Paths stack
  global myPathName, myPath, myHistory, myPathCard, myMetaInfo, myDelim, myPalette,
PathStatus, openPathStatus
  put "Select the Path that you want to follow..." into listPrompt
  push cd
  pop cd into myCurrCd
  go to stack myPathName

  -- get list of possible paths (i.e. cards in the stack)
  put empty into myPathList
  go to cd 2 of this stack
  repeat with n = 2 to the number of cds
    put the short name of this cd & return after myPathList
    go next cd
  end repeat

  -- now display the list of available paths and select one.
  -- Only allow "New" path if the palette is in Author mode
  put "  New  " into myButton
  repeat forever
    if myPalette = "Author" then
      put XScrollBox(1,listPrompt,myPathList, myButton) into tempPathCard
    else
      put XScrollBox(1,listPrompt,myPathList) into tempPathCard
    end if
    if tempPathCard is empty then  -- "Cancel" was pressed
      -- do nothing and exit repeat loop
      put "cancel" into openPathStatus
      exit repeat
    else
      if item 1 of tempPathCard = myButton then  -- create a new path
        ask "Enter the new path name" with "New Path"
        if the result = "Cancel" then
          put "cancel" into openPathStatus
          exit repeat
```

```
        else
          if it is empty then -- do nothing and re-enter loop
          else
            put it into myPathCard
            go last cd
            doMenu "New Card"
            set the name of this cd to myPathCard
            put empty into myPath
            put myDelim into myMetaInfo
            put myDelim into bg fld "metainformation"
            put empty into myHistory
            put "New" into PathStatus
            put "New" into openPathStatus
            exit repeat
          end if
        end if
      else  -- OK was pressed and path was selected
        if item 2 of tempPathCard is not empty then  -- path was selected
          put item 2 of tempPathCard into myPathCard
          put bg fld "Path" of cd myPathCard into myPath
          put bg fld "metaInformation" of cd myPathCard into myMetaInfo
          put bg fld History of cd myPathCard into myHistory
          put "Current" into openPathStatus
          exit repeat
        end if
      end if
    end if
  end repeat
end getPath

on goToNode nodeNumber
  -- this goes to the node in the path that is identified by the parameter nodeNumber
  global myPath
  put item 1 of line NodeNumber of myPath into myCardID -- get node ID
  put item 2 of line NodeNumber of myPath into myStack  -- get stack name
  go cd id myCardID of stack myStack
  if myCardID <> the short id of this cd then
```

```
    answer "Card id" && myCardID && "in stack" && myStack && "no longer exists." with
"Ok"
  end if
end goToNode

on updateInfo newNodeNum, myNodeNum
  -- This updates the meta-information window when a new node is accessed
  -- so that it displays the meta-information for the right node.
  global myMetaInfo, myInfoWindow, infoEditStatus
  if there is a window myInfoWindow then

    if infoEditStatus = "TRUE" then
      put the text of window myInfoWindow into myInfo
      insertInfo myNodeNum, myInfo
      set the lockText of window myInfoWindow to true
      put "FALSE" into infoEditStatus
    end if

    put parseInfo(newNodeNum) into myInfo
    -- if myInfo is empty then
    -- closeInfoWindow     -- may be better to display the blank window
    -- and let the user close it. The user should have control.
    -- else
    set text of window myInfoWindow to myInfo
    -- end if
  end if
end updateInfo

on metaInfo
  -- parse the metaInformation field to put meta-information window
  -- on the screen, using an external window
  global currNodeNum, myMetaInfo, myInfoWindow, myInfoRect, myPalette
  lock screen
  if currNodeNum = 0 then
    answer "Can not add or display information until you have added a node." with "Ok"
    exit metaInfo
  end if
```

```
  put parseInfo(currNodeNum) into myInfo
  if myInfo is empty then   -- can add some meta-info
    if myPalette = "Author" then
      answer "Add some information for this node?" with "Cancel", "Add"
      if it = "Cancel" then
        -- do nothing
      else
        AddMetaInfo currNodeNum
      end if
    else
      answer "There is no meta-information for this node." with "Ok"
    end if
  else
    goToNode currNodeNum
    textoid myInfoWindow, myInfo, myInfoRect  -- courtesy F. Rinaldi
  end if
end metaInfo

on AddMetaInfo nodeNumber
  -- This displays the meta-information window in edit mode so an author
  -- may enter meta-information for a node.
  global myMetaInfo, infoEditStatus, myInfoRect, myInfoWindow, currNodeNum
  lock screen
  put "TRUE" into infoEditStatus
  -- put empty into myInfo
  -- changed this in case we want to edit the info.
  put parseInfo(currNodeNum) into myInfo
  textoid myInfoWindow, myInfo, myInfoRect
  set the lockText of window myInfoWindow to false
  unlock screen
end AddMetaInfo

on clicInText What,Where,Start,Stop,Font,Size,Style,Clr
  -- This close the meta-information window. The clicInText message
  -- is sent when the external window, created using the Textoid XCMD,
  -- is clicked in using the mouse. We could, perhaps, do other processing
  -- if desired, such as processing the text chunk clicked in for use as
```

```
  -- a hypertext link.
  closeInfoWindow
end clicInText

on closeTextoid which
  -- This closes the meta-information window. First we must check
  --     that this message is sent from the meta-information window. If
  --     it isn't then we can pass it on.
  -- If the meta-information is being edited then we must save it
  --     for this node.
  global currNodeNum, myInfoWindow, infoEditStatus
  if which = myInfoWindow then
    if infoEditStatus = "TRUE" then
      get the text of window which
      put it into myInfo
      insertInfo currNodeNum, myInfo
      put "FALSE" into infoEditStatus
    end if
  end if
  saveInfoRect
end closeTextoid

on insertInfo nodeNumber, myText
  -- This inserts meta-information into the correct position in the
  --     meta-information structure. It must parse the structure and count
  --     the occurences of the delimiter to determine the correct position.
  global myMetaInfo, myDelim
  if NodeNumber = 0 then
    put myDelim & return & myDelim into myMetaInfo
  else
    put nodeNumber - 1 into prevNode
    put 0 into x
    put 0 into thisNode
    repeat forever
      put x + 1 into x
      if (line x of myMetaInfo) = myDelim then
        if thisNode = prevNode then
```

```
          put return & myText after line x of myMetaInfo
          exit repeat
        else
          put thisNode + 1 into thisNode
        end if
      end if
    end repeat
  end if
end insertInfo

on closeInfoWindow
  global myInfoWindow
  if there is a window myInfoWindow then
    close window myInfoWindow
  end if
end closeInfoWindow

on saveInfoRect
  -- this saves the last position of the meta-information window
  -- so that it can be re-opened in the same position as last time.
  global myInfoWindow, myInfoRect
  put the globalRect of window myInfoWindow into myInfoRect
end saveInfoRect

function parseInfo nodeNumber
  -- This function returns text which corresponds to a node number.
  -- The text for all nodes is in the memory variable myMetaInfo.
  -- The delimiters separating each node's text is the word "endNode".
  --
  -- This function returns an empty string if there is no information
  -- for the current node.
  global myMetaInfo, myDelim
  if nodeNumber = 0 then return empty
  put 20 into infoLimit
  put empty into myInfo
  put nodeNumber - 1 into prevNode
  put 0 into x
```

```
  put 0 into targetNode
  repeat forever
    put x + 1 into x
    if (line x of myMetaInfo) = myDelim then
      if targetNode <> prevNode then
        put targetNode + 1 into targetNode
      else
        repeat forever
          put x + 1 into x
          if line x of myMetaInfo <> myDelim then
            put line x of myMetaInfo & return after myInfo
          else
            exit repeat
          end if
        end repeat
        exit repeat
      end if
    end if
  end repeat
  return myInfo
end parseInfo

on deleteMetaInfo nodeNum
  -- Delete the metainformation for a node.
  -- Note that we must also delete the 2nd delimiter.
  global myMetaInfo, myDelim
  put 1 into lineNum
  put 0 into myNode
  repeat forever
    if line lineNum of myMetaInfo = myDelim then
      put myNode + 1 into myNode
      if myNode = nodeNum then
        put lineNum + 1 into lineNum
        repeat forever
          if line lineNum of myMetaInfo <> myDelim then
            delete line lineNum of myMetaInfo
          else
```

```
          exit repeat
        end if
      end repeat
      delete line lineNum of myMetaInfo  -- delete the 2nd delimiter
      exit repeat
    end if
  end if
  put lineNum + 1 into lineNum
 end repeat
end deleteMetaInfo

on nextPath
  -- this takes the user to the next node in the path
  -- if the user is at the last node then display a message
  -- add the previous node to the history list

  -- myPath contains the contents of the current Path
  -- myHistory contains the current user's history
  -- currNodeNum contains the line number of the current Node
  -- startTicks contains the start time that this node was begun
  global currNodeNum, myPath, startTicks
  lock screen
  set lockMessages to true
  put "visual effect wipe left" into visualEffect
  if currNodeNum > 0 then
    -- update history list
    AddHistory (the short id of this cd), (the short name of this stack)
  end if
  if currNodeNum = the number of lines in myPath then
    answer "End of path reached." with "Ok"
  else
    -- now go to next node
    put currNodeNum + 1 into currNodeNum  -- increments path node counter
    goToNode currNodeNum
    updateInfo currNodeNum, currNodeNum -1
  end if
  put the ticks into startTicks
```

```
  unlock screen with visualEffect
end nextPath

on prevPath
  -- This takes the user to the previous node in the path.
  -- If the user is at the first node then display a message.
  -- Add the previous node to the history list

  -- myPath contains the contents of the current Path
  -- myHistory contains the current user's history
  -- currNodeNum contains the line number of the current Node
  -- startTicks contains the start time that this node was begun

  global currNodeNum, startTicks
  lock screen
  set lockMessages to true
  put "visual effect wipe right" into visualEffect
  if currNodeNum > 0 then
    -- update history list
    AddHistory (the short id of this cd), (the short name of this stack)
  end if
  if currNodeNum > 1 then
    -- now go to prev node
    put currNodeNum - 1 into currNodeNum  -- increments path node counter
    goToNode currNodeNum
    updateInfo currNodeNum, currNodeNum +1
  else
    answer "Already at the beginning of the path." with "Ok"
  end if
  put the ticks into startTicks
  unlock screen with visualEffect
end prevPath

on lastPath
  -- Jump back to the current node in the path.
  -- This should be used when a user has made a diversion and wants
  -- to get back to the current node directly.
```

```
  global currNodeNum
  lock screen
  set lockMessages to true
  put "visual effect iris close" into visualEffect
  AddHistory (the short id of this cd), (the short name of this stack)
  goToNode currNodeNum
  unlock screen with visualEffect
end lastPath

on deletePath
  -- Delete a node from the path.
  -- Go to the node to be deleted and confirm deletion first.
  -- If confirmed then
  --    delete this node from the path
  --    do associated details such as decrementing the current node number
  --    then go to the node after (or before, if at last node) the deleted node
  global currNodeNum, myPath
  lock screen
  set lockMessages to true
  goToNode currNodeNum
  unlock screen          -- to update the screen for the user
  lock screen
  answer "Are you sure you want to delete the current node?" ¬
  with "Cancel", "Delete"
  if it is "Cancel" then exit deletePath
  if it is "Delete" then
    delete line currNodeNum of myPath
    deleteMetaInfo currNodeNum
    -- check if the node we deleted was the last node.
    -- if it was then go to the previous node
    if currNodeNum > the number of lines in myPath then
      put currNodeNum - 1 into currNodeNum
    end if
    goToNode currNodeNum        -- this will take the user to the next
    -- node in the path or the previous one if the last node was deleted
    end deletePath
```

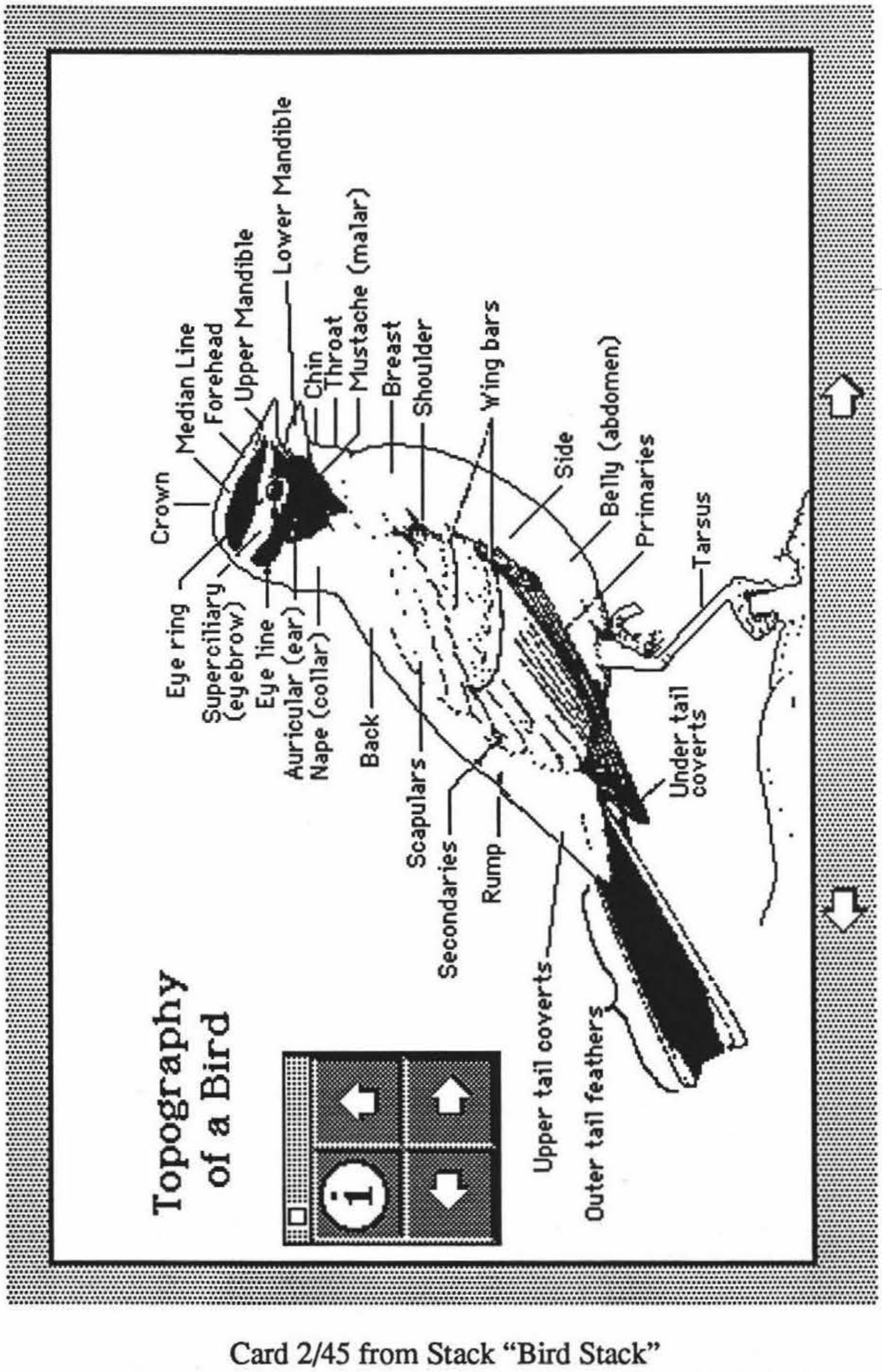
```
on AddPath
  -- This adds a new node to the current Path.
  -- Add the previous node to the history list.

  -- We might want to search the path structure to see if the current
  -- node is in the path already - although not really necessary
  -- as we might want to visit the same node twice.

  --    myPath contains the contents of the current Path
  --    myHistory contains the current user's history
  --    currNodeNum contains the line number of the current Node
  --    startTicks contains the start time that this node was begun

  global myPath, myHistory, currNodeNum, startTicks, myDelim, myInfoWindow
  if there is a window myInfoWindow then close window myInfoWindow
  put the short id of this card into myCard
  put the short name of this stack into myStack
  if currNodeNum = 0 then
    put myCard & "," & myStack & return into myPath
    put myDelim into myMetaInfo
  else
    if currNodeNum >= the number of lines in myPath then
      put myCard & "," & myStack & return after myPath
    else
      -- insert node
      put return & myCard & "," & myStack after line currNodeNum of myPath
    end if
  end if
  AddHistory (the first item of line currNodeNum of myPath), ¬
           (the second item of line currNodeNum of myPath)
  put currNodeNum + 1 into currNodeNum
  InsertInfo currNodeNum, myDelim      -- this makes room for the new node'e meta-info
end AddPath


on savePath
```

```
  -- This will save the current path and it's history.
  -- We can create a new path card or overwrite the current one.

  global myPathName, myPath, myHistory, myPathCard, myMetaInfo, PathStatus
  -- overwrite existing path?
  if PathStatus <> "New" then
    answer "Overwrite current path" && myPathCard & "?" with "Cancel", ¬
                                                        "Overwrite", "New"
  else
    answer "Save current path" && myPathCard & "?" with "Cancel", "Save"
  end if
  lock screen
  if it is "Cancel" then exit savePath
  if it is "New" then
    go to last cd of stack myPathName
    doMenu "New Card"
    ask "Please Enter the name of this new path..." with "New Path"
    if ((it is empty) or (the result = "Cancel")) then
      exit savePath
    else
      set the name of this cd to it
      put it into bg fld "Path Name"
      put it into myPathCard
    end if
  else  -- it is "Overwrite" or "Save"
    go to cd myPathCard of stack myPathName
  end if
  put myPath into bg fld "Path"
  put myMetaInfo into bg fld "MetaInformation"
  put myHistory into bg fld "History"
  -- now path has been saved then we must overwrite it next time
  put empty into PathStatus
end savePath

on AddHistory myCard, myStack
  -- This saves the current card details to the History structure.
  global myHistory, startTicks
```

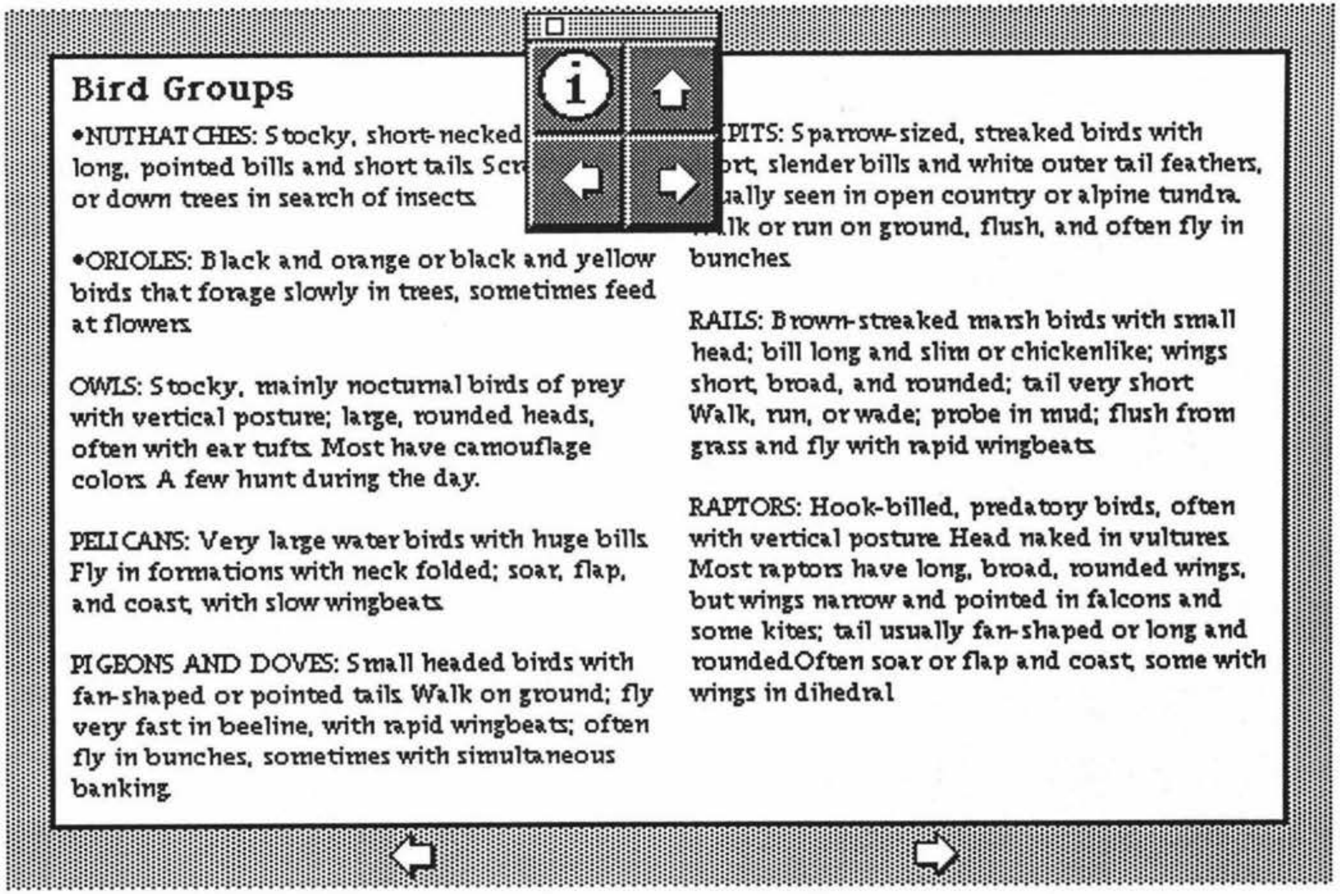
```
  put the ticks into endTicks
  put (endTicks - startTicks) / 60 into elapsedTime
  put the short time into myTime
  put myCard & "," & myStack & "," & myTime & "," & elapsedTime ¬
                                                  & return after myHistory
  put the ticks into startTicks
end AddHistory

on createPathMenu
  if there is a menu "Paths" then exit createPathMenu
  create menu "Paths"
  put createPathMenuItems() into menu "Paths" with menuMsg createPathMenuMsgs()
  disable menuItem 5 of menu "Paths"
  disable menuItem 8 of menu "Paths"
end createPathMenu

function createPathMenuItems
  return "Next" & return & "Previous" & return & "Information" & return & ¬
  "Jump back" & return & "-" & return & "Add" & return & "Delete" & return & ¬
  "-" & return & "Open..." & return & "Save" & return
end createPathMenuItems

function createPathMenuMsgs
  return "NextPath" & return & "PrevPath" & return & "metaInfo" & return & ¬
  "lastPath" & return & "" & return & "addPath" & return & "deletePath" & return & -
  "" & return & "openPath" & return & "savePath"
end createPathMenuMsgs

on UpdateMenu myItems, itemStatus
  if there is not a menu "Paths" then exit UpdateMenu
  repeat with x = 1 to the number of items in myItems
    put item x of myItems into myMenuItem
    do itemStatus && "menuitem" && myMenuItem && "of menu Paths"
  end repeat
end UpdateMenu
```

Cards on the Guided Path

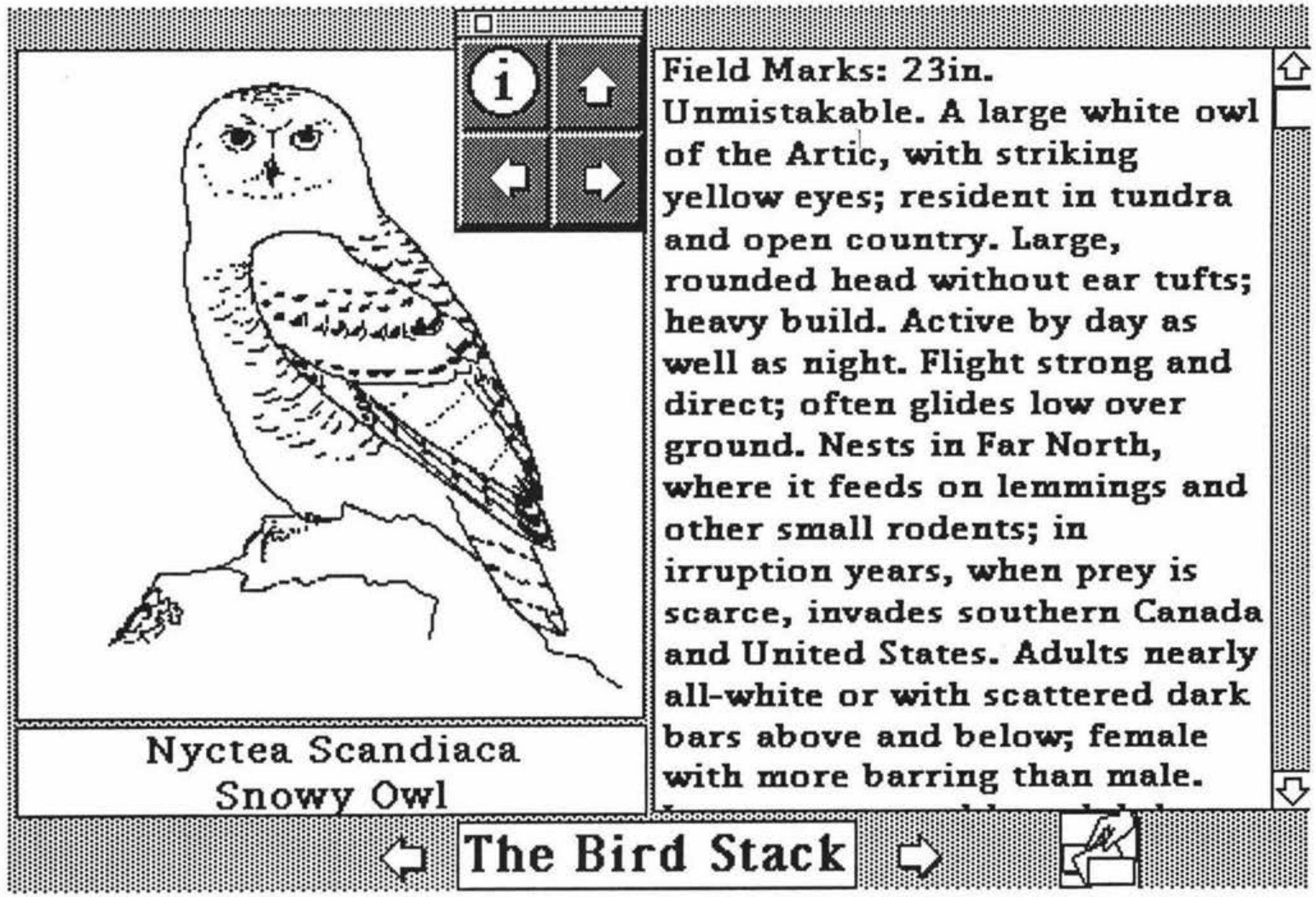
This section contains the cards that were on the path. Below each card is a description of its location within the stack in which it appears.



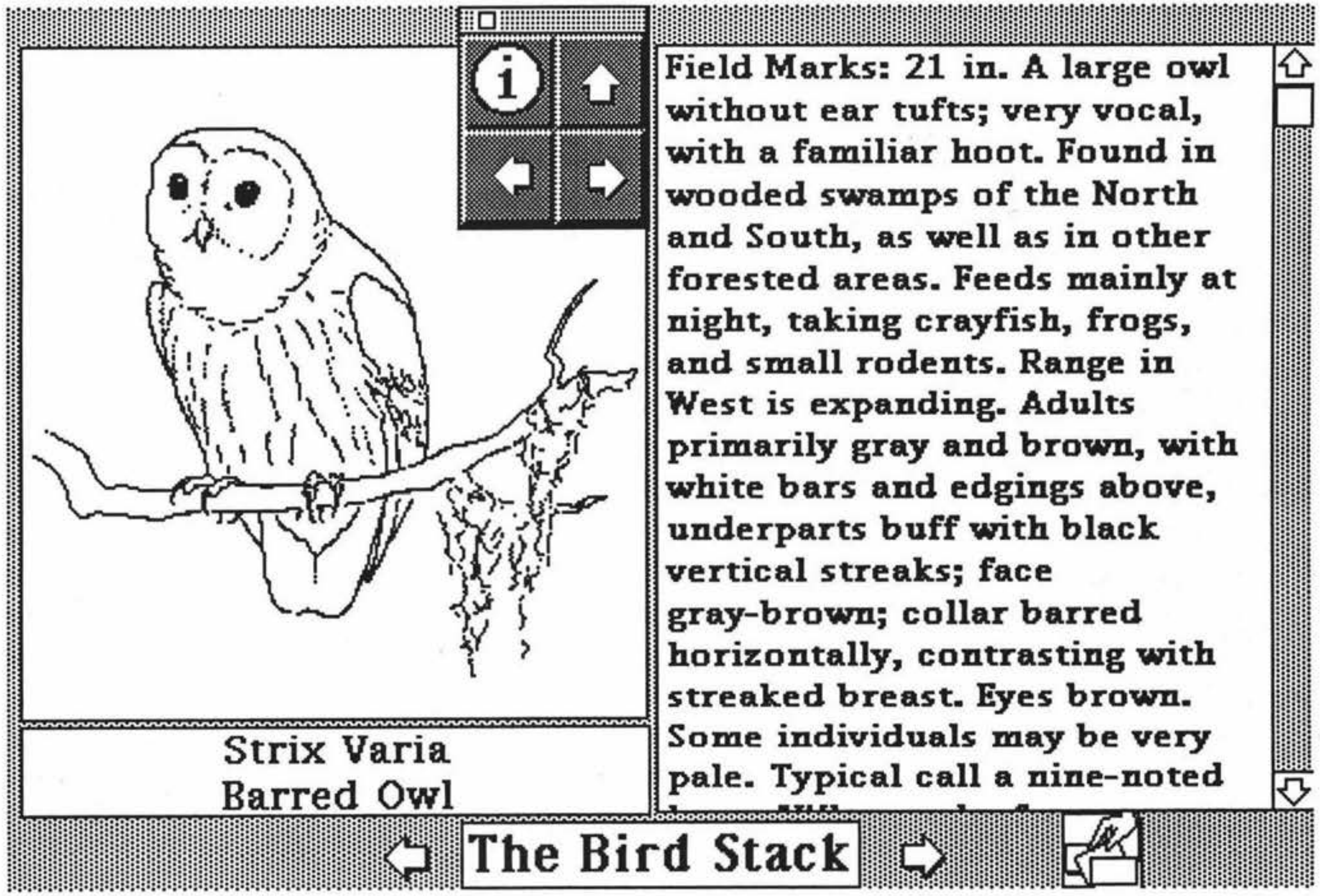
Card 2/45 from Stack "Bird Stack"



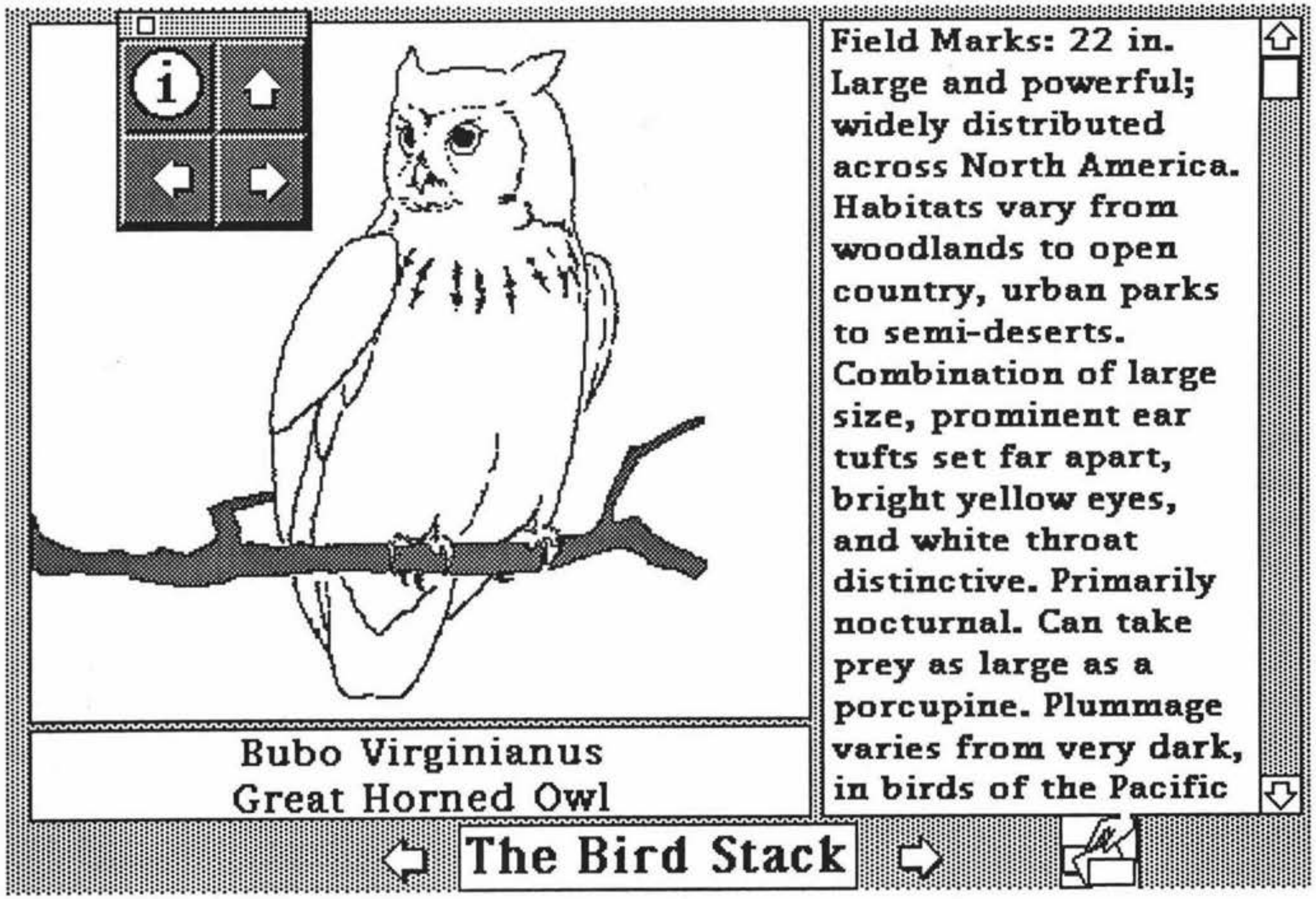
Card 7/45 from Stack "Bird Stack"



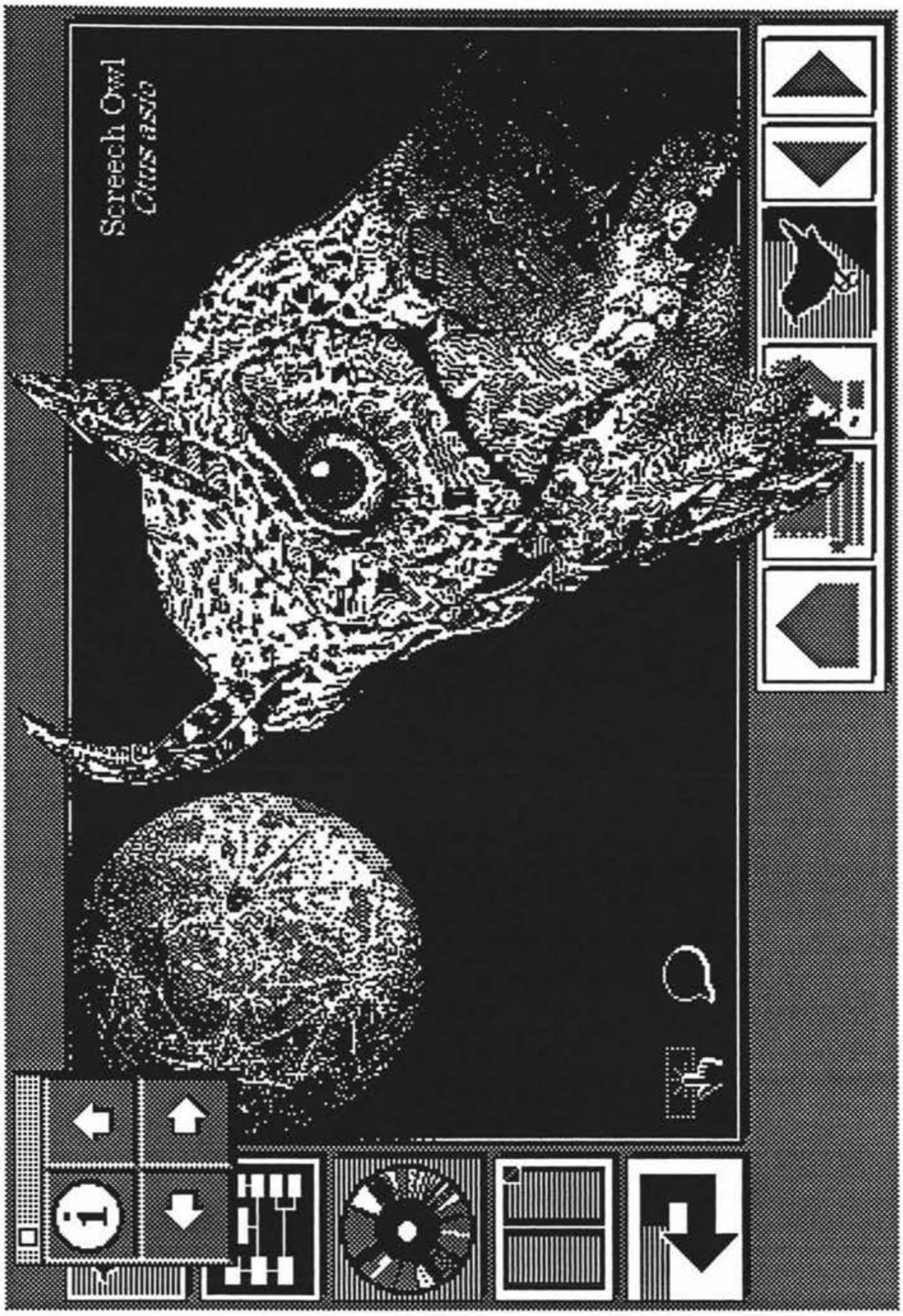
Card 29/45 from Stack "Bird Stack"



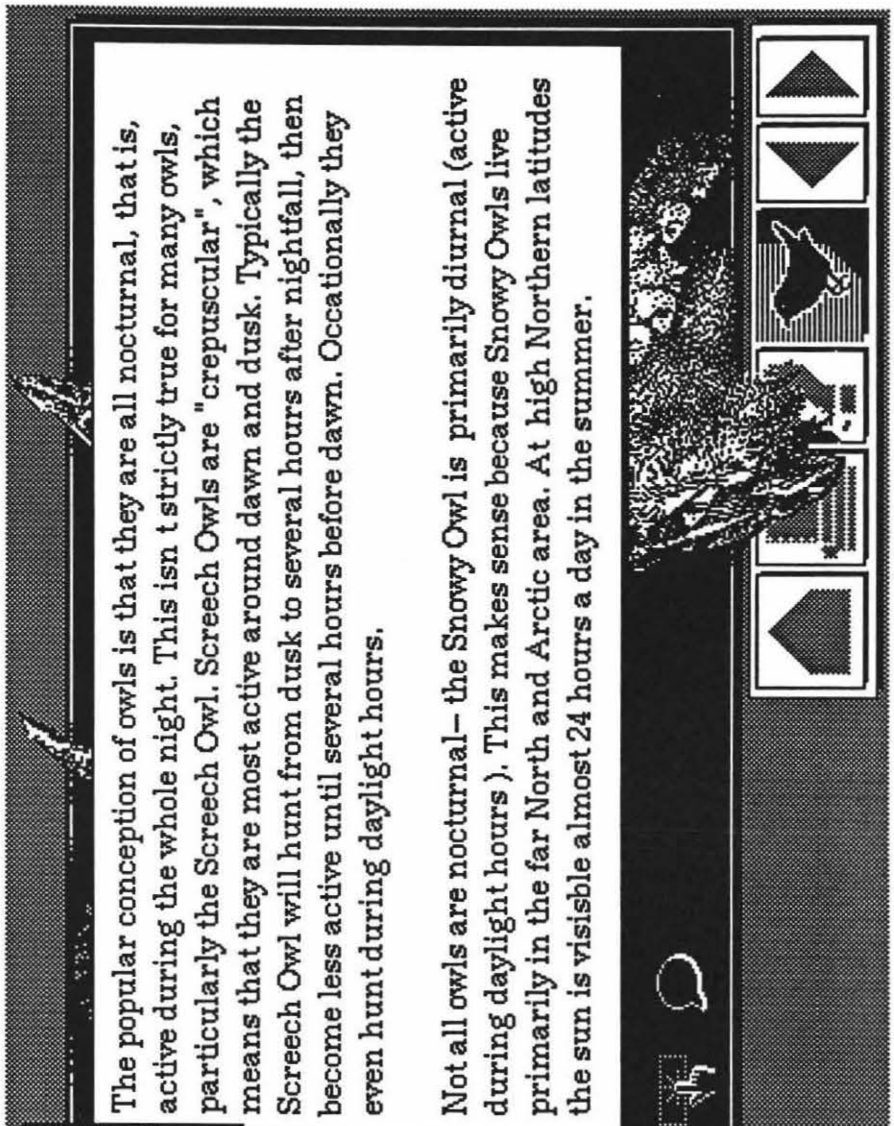
Card 30/45 from Stack "Bird Stack"



Card 31/45 from Stack "Bird Stack"

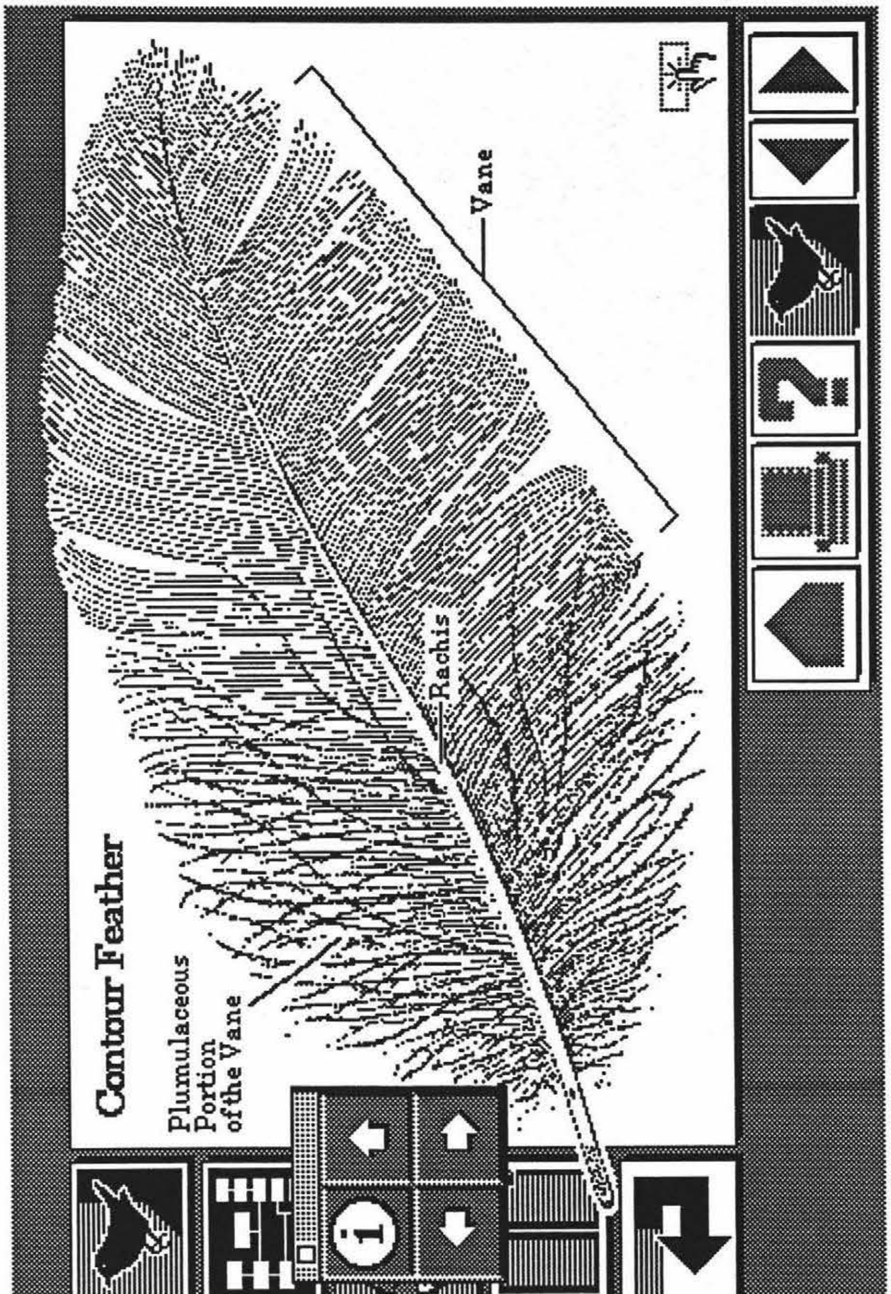


Card 12/47 from Stack "Bird Anatomy"

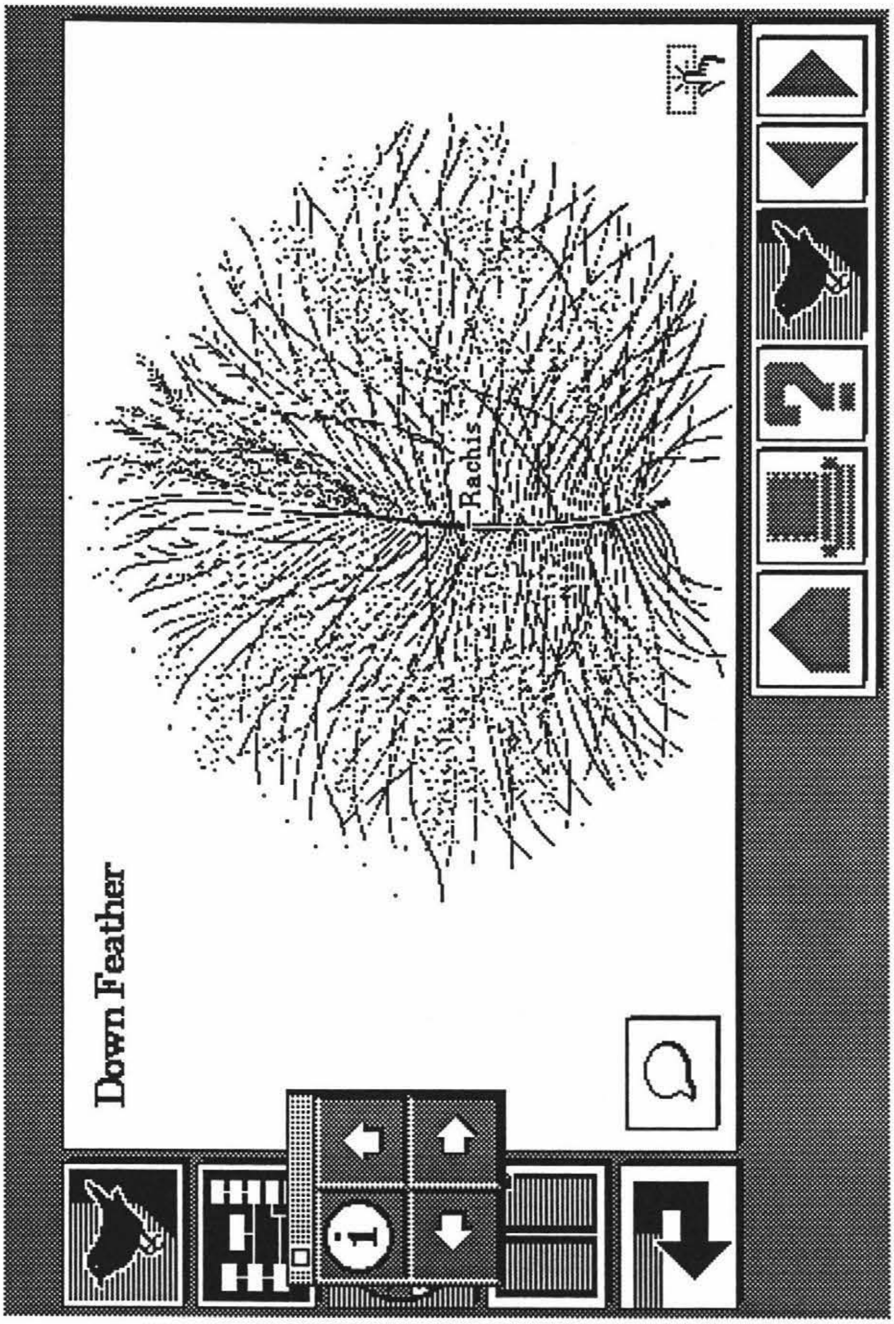


Card 12/47 from Stack "Bird Anatomy"

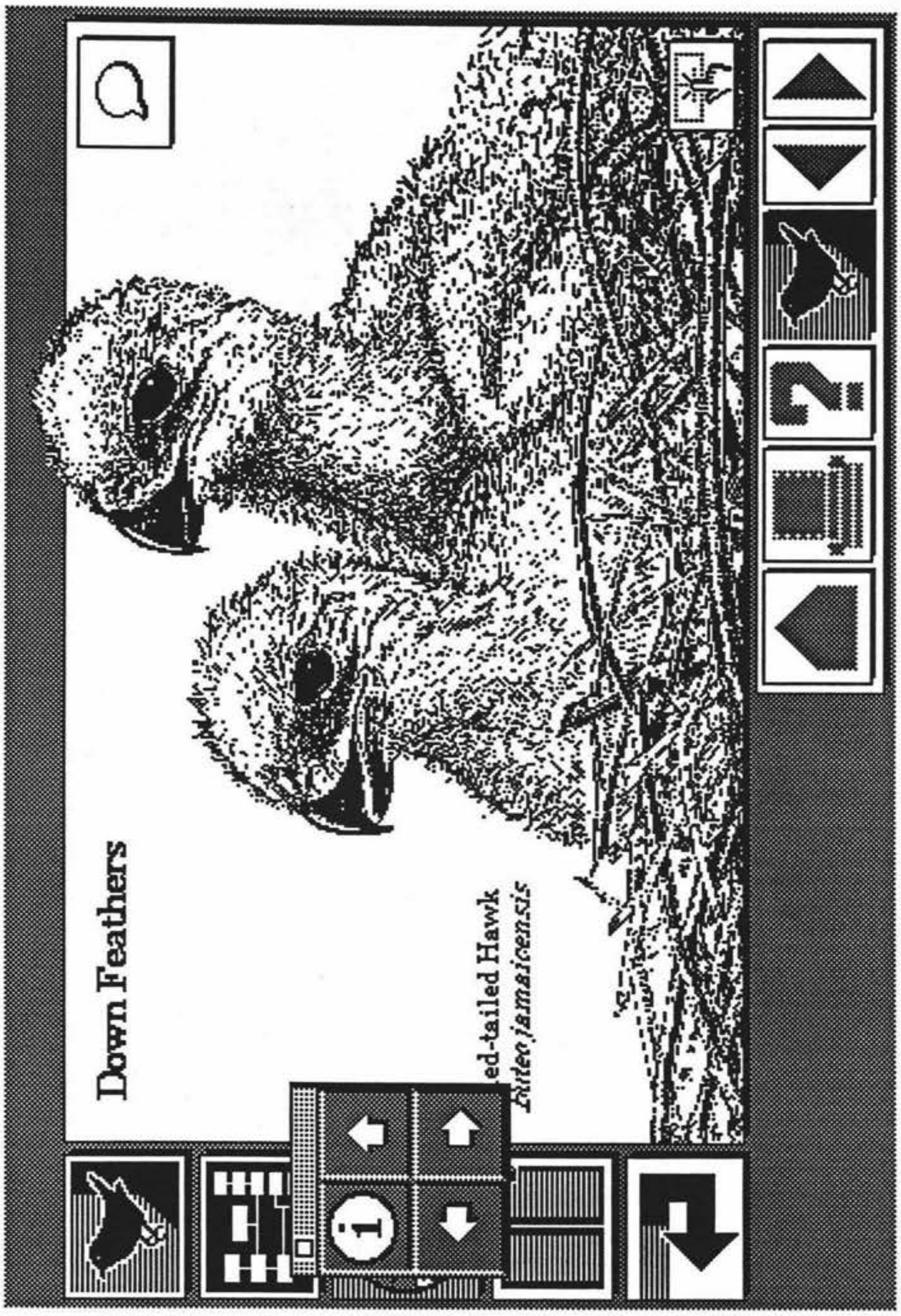
This card shows the result of clicking on the small information bubble icon near the bottom left of the card. A pop-up field appears that provides general information about the Screech Owl. The pop-up field overlaps much of the card display and can be hidden by clicking on the field itself — this is a common convention in HyperCard stacks.



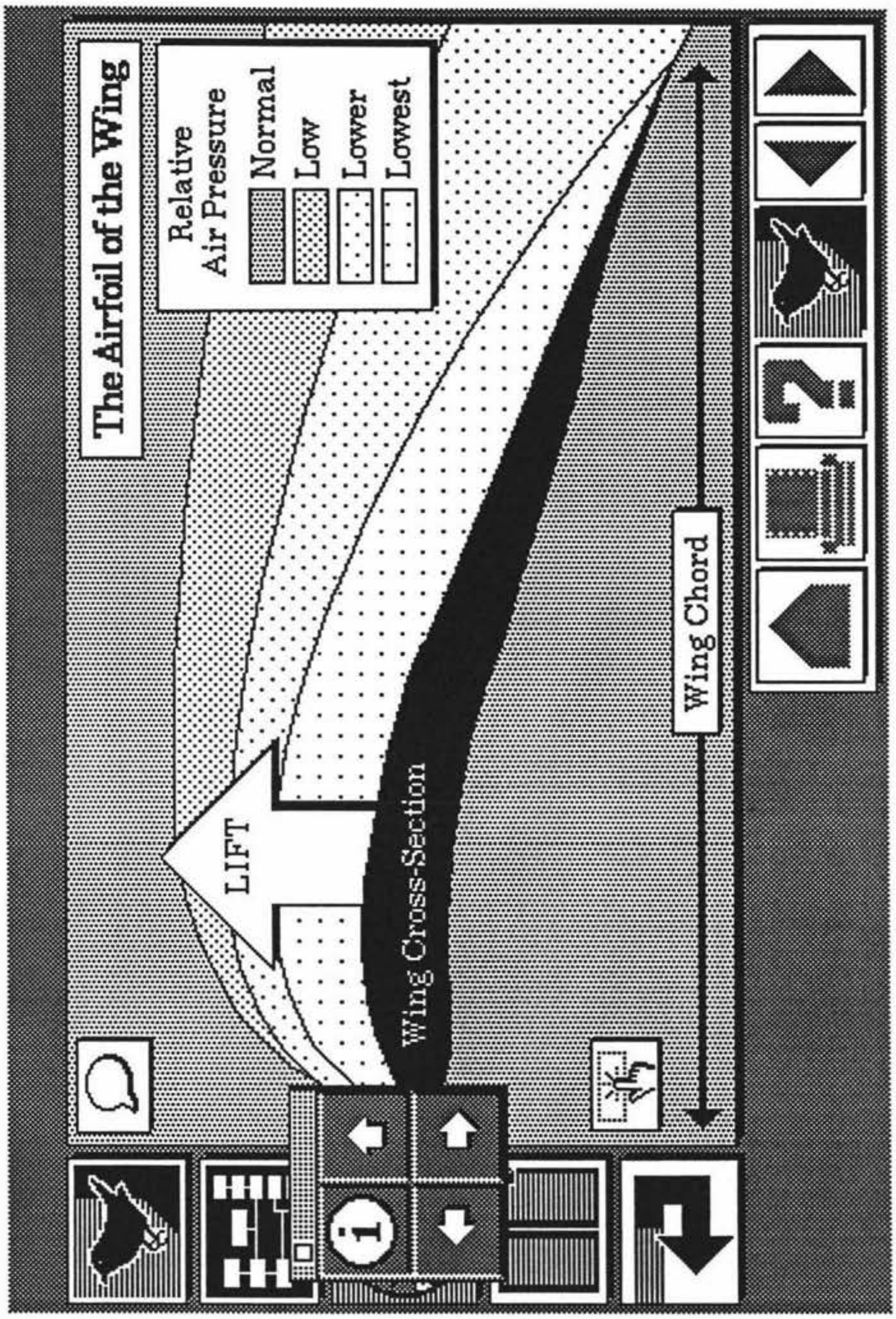
Card 19/47 from Stack "Bird Anatomy"



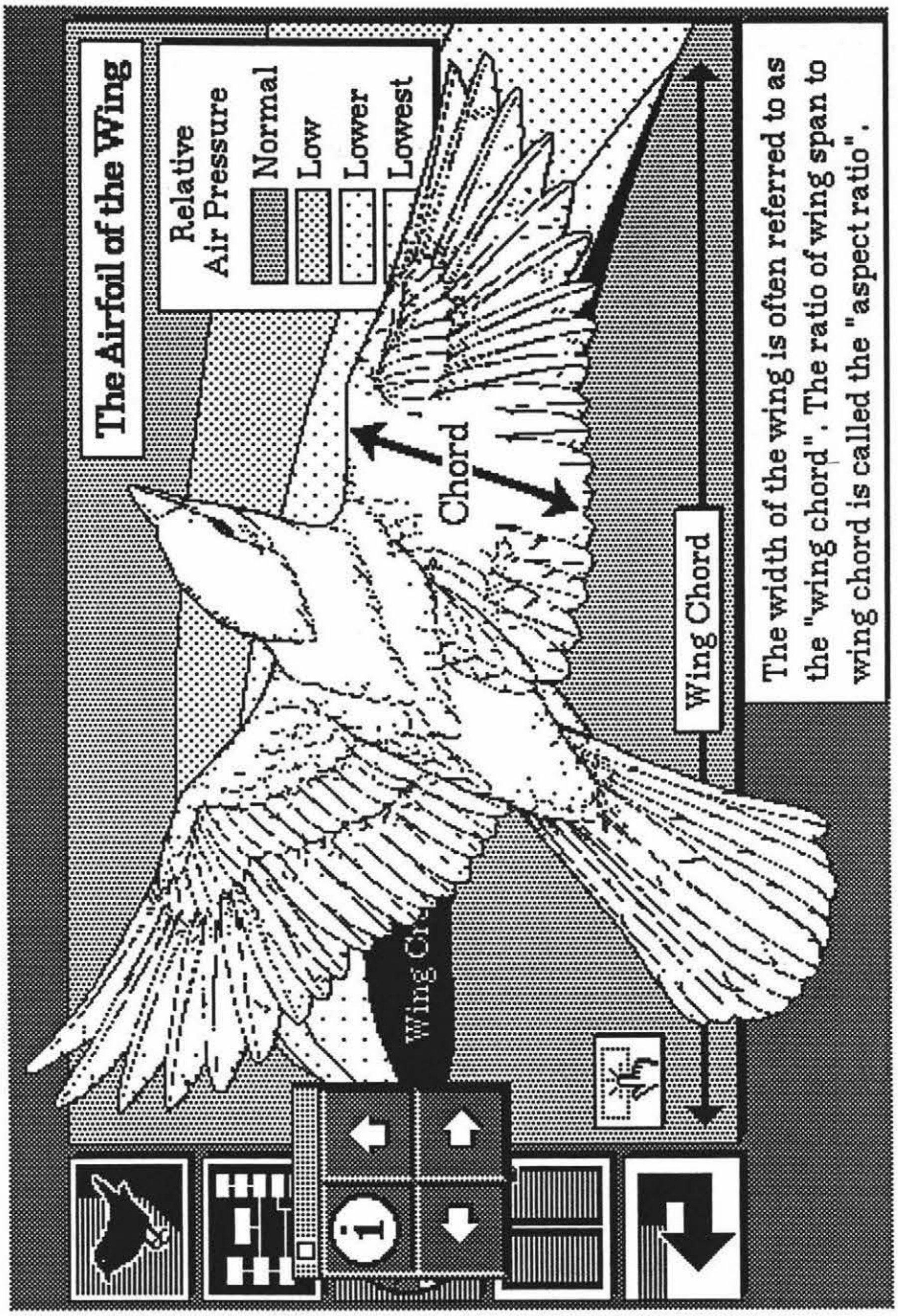
Card 22/47 from Stack "Bird Anatomy"



Card 23/47 from Stack "Bird Anatomy"



Card 33/47 from Stack "Bird Anatomy"



Card 37/47 from Stack "Bird Anatomy"