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, exact details of how to use 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.

Ray 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 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:

Function	Command-key	Description
Back	Command-~	Go to the previous card. This can be a card in another stack if a jump was made to the current card.
Home	Command-H	Go to the Home stack.
Help	Command-?	Go to the Hypercard Help stack for information about Hypercard.

Recent	Command-R	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	Command-1	Go to the first card in this stack
Prev	Command-2	Go to the card before the current card in this stack.
Next	Command-3	Go to the card after the current card in this stack.
Last	Command-4	Go to the last card in this stack.
Find…	Command-F	Brings up the find dialog. Allows the user to search for a text string in a field.
Message	Command-M	Brings up the message box and allows the user to enter a Hypertalk expression.
Scroll	Command-E	Allows the user to reduce, enlarge, or scroll the current stack window.
Next Window	Command-L	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*

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.

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 take the user to further explanation of the graphic.

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.

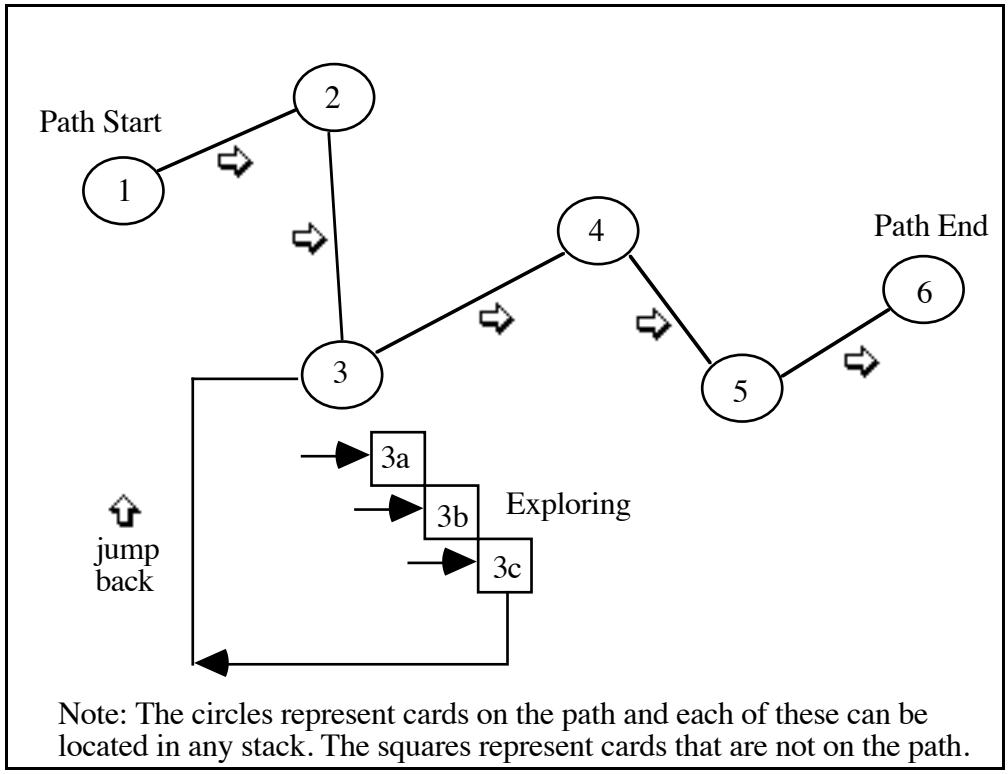

Figure 4.1 *Path and off-the-path exploration.*

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:

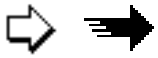	go to the next card
	go to the previous card
	go to the card that a hypertext jump was last made from (not necessarily the previous card)
	go to the home card
	go to the last card
	go to the first card
	display help

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.

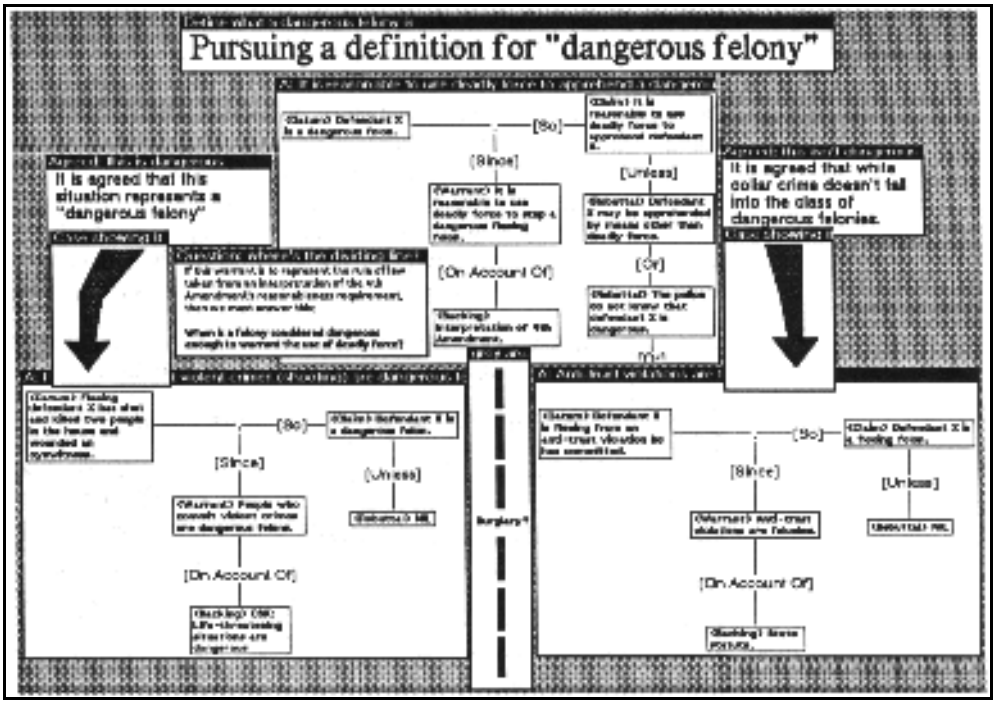

Figure 4.2 *NoteCards fragmentation, and arrows used as meta-information devices to direct attention.*

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.

Working Details — How to use it

The paths facility provides different functions for two groups of users which I have 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.

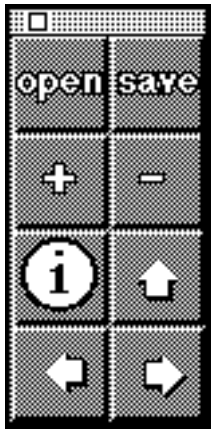

Figure 4.3 *Author Palette*

3.1 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 4.4. To create a new path, the author should click on the New button. After doing this, figure 4.5 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.

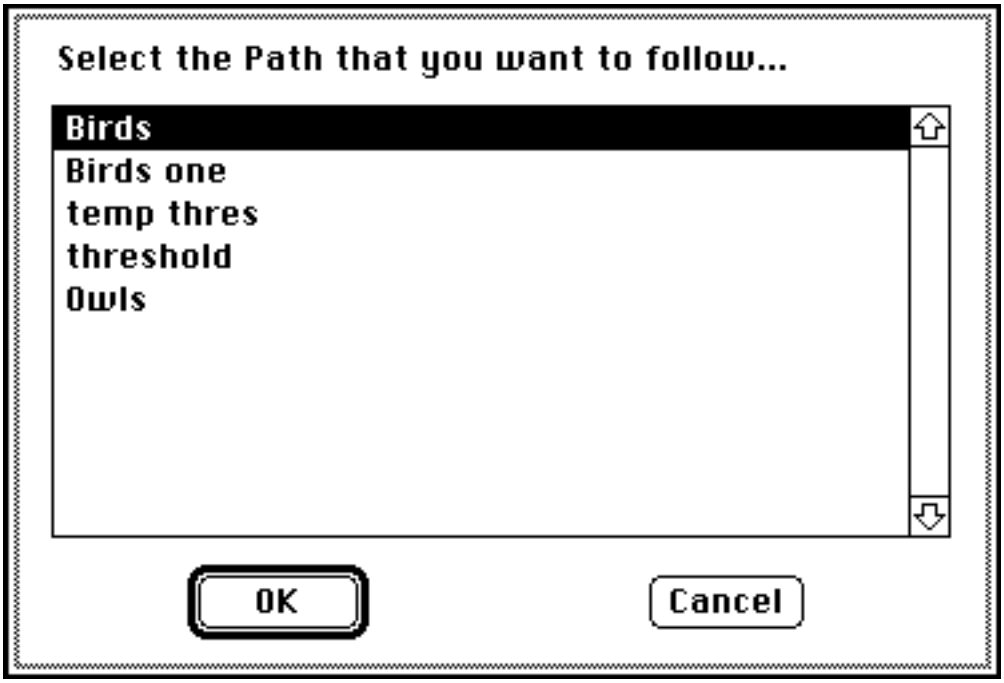

Figure 4.4 *Open Path Dialog*

Figure 4.5 *Name Path*

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.

3.2 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 4.4 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.

3.3 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.

3.4 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 4.6. 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.

Figure 4.6 *Confirm node deletion dialog*

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 4.6.

3.5 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 4.7. Three options are available: Cancel, Overwrite, or New.

Figure 4.7 *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 4.5. 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.

3.6 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.

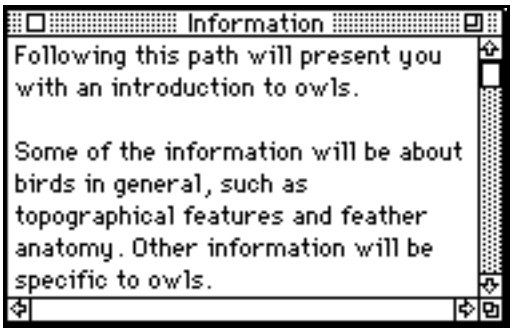

Figure 4.8 *Meta-information Window*

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.

3.7 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.

3.8 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.

3.9 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.

4. 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.

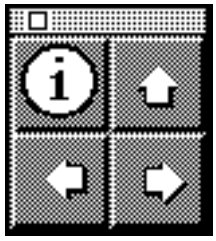

Figure 4.9 *Path palette for students*

4.1 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.