

Rochester Institute of Technology

## RIT Digital Institutional Repository

---

### Theses

---

1991

## HyperManual a hypermedia construction and browsing system in Smalltalk/V Mac

James Tracy Burdick

Follow this and additional works at: <https://repository.rit.edu/theses>

---

### Recommended Citation

Burdick, James Tracy, "HyperManual a hypermedia construction and browsing system in Smalltalk/V Mac" (1991). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact [repository@rit.edu](mailto:repository@rit.edu).

Rochester Institute of Technology  
Department of Computer Science

**HyperManual**  
**A Hypermedia Construction and Browsing System**  
**in Smalltalk/V Mac**

by James Tracy Burdick

A thesis, submitted to  
The Faculty of the Department of Computer Science  
in partial fulfillment of the requirements for the degree of  
Master of Science in Computer Science

Approved by:        Robert T. Gayvert

John. A. Biles

Peter G. Anderson

# HyperManual

I, James Tracy Burdick  
hereby deny permission for the Wallace Memorial Library, of RIT, to  
reproduce my thesis in whole or in part.

# HyperManual

## Abstract

The process of locating needed information in a paper document is often frustrating and time consuming. The information may be available but, since the basic access method is often linear, the information is not readily available. Hypermedia systems offer a computer-based mechanism that can simplify the search process and increase the ratio of useful information found to time expended. Such systems also support the creation of alternate views on the same data base of information.

This thesis describes some current hypertext systems and the overall structure of the HyperManual system. This system includes both the tools needed to create and traverse a web of links within a database of documents, and an environment that supports these link construction and information browsing activities. The system differs from other hypermedia systems in that it provides a variety of modes of access to the documents in a hyperdocument. The HyperManual system was written in Smalltalk/V for the Macintosh\* .

### Key Words and Phrases

Hypertext

HyperMedia

Graphical User Interfaces

Smalltalk

### Computing Review Subject Codes

H.5.1: Multimedia Information Systems

H.5.2: User Interfaces

I.7.2: Document preparation

---

\* Smalltalk/V is a registered trademark of Digitalk, Inc.

# HyperManual

## Table of Contents

I. Introduction.....	1
Overview of Thesis.....	1
II. Background.....	2
History of Hypertext .....	2
Current Systems .....	3
III. HyperManual Implementation .....	13
Design Issues.....	13
Structural Issues.....	13
Interface Issues .....	13
HyperManual System Description .....	14
Hyperdocument Structure .....	14
Browsing Mode .....	14
Linking Mode.....	15
Uniform User Interface.....	15
Minimization of System Complexity.....	15
Annotations .....	17
Class Hierarchy .....	17
Dispatchers.....	17
Panels.....	18
Models.....	18
Chronology.....	23
IV. Results.....	26
Integration .....	26
Modifiability.....	26
Criticism of Final Product .....	26
Product Overview.....	27
V. Conclusions .....	29
The Hypertext Concept.....	29
The HyperManual System.....	29
Future Extensions.....	30
Bibliography.....	33
Appendix A. User Manual.....	35
Appendix B. Class Hierarchy .....	67
Appendix C. Glossary.....	70

# HyperManual

## Table of Contents

### Figures

Fig. 1	--	Intermedia: Local Map.....	7
Fig. 2	--	Table of Contents .....	21
Fig. 3	--	Index.....	22
Fig. 4	--	Annotation Browser .....	23
Fig. 5	--	HyperManual Initialize .....	37
Fig. 6	--	Tool Palette .....	38
Fig. 7	--	Table of Contents .....	39
Fig. 8	--	Index Window.....	41
Fig. 9	--	Annotation Browser .....	43
Fig. 10	--	Document Window .....	46
Fig. 11	--	Destination List Window.....	47
Fig. 12	--	Note Window.....	48
Fig. 13	--	New Annotation Window.....	48
Fig. 14	--	Annotation Window .....	49
Fig. 15	--	HyperText Menu .....	51
Fig. 16	--	Icon Ribbon.....	51
Fig. 17	--	Change Preferences Dialog Box.....	55
Fig. 18	--	Link Menu .....	56
Fig. 19	--	Source Selection Overlap Dialog Box .....	57
Fig. 20	--	Save Link Dialog Box .....	59
Fig. 21	--	Link Icon Ribbon .....	60
Fig. 22	--	Info Dialog Box .....	62
Fig. 23	--	File Menu .....	62
Fig. 24	--	Index Menu .....	64
Fig. 25	--	Index Icon Ribbon .....	65

## **I. Introduction**

A fundamental problem with large technical manuals and instructional books is finding the the needed information quickly and easily. Paper documents may include a table of contents, list of figures, and an index, but each of these tends to be incomplete since the space allotted to these access tools is only a small portion of the entire document. It can be very difficult to locate the desired information and when it is located, there rarely is an easy way to access related information. Following cross references can be a tedious and time consuming task of locating the correct volume, entry, and (finally) the relevant portion of the referenced document. Hypermedia systems offer a computer-based mechanism that can reduce the drudgery involved in finding information. They also offer a framework that can support both non-sequential associations among data and non-sequential access to data in addition to standard sequential data access.

This thesis describes the design and general implementation of a hypermedia system called HyperManual. The name HyperManual was derived from the original purpose of the project ,which was to design a system that could be used to create and browse hypermedia versions of technical manuals. The system includes both the tools needed to create and traverse a web of links within a database of documents and an environment that supports these link construction and information browsing activities. The HyperManual system was written in Smalltalk/V for the Macintosh.\*

### **Overview of Thesis**

The introduction, section I, describes the general problem of finding information in complex documents and gives a brief description of a partial solution. The background, section II, sketches the history of hypertext and discusses some current products with a particular emphasis on Intermedia from Brown University and Guide from Owl International. Section III describes the design and implementation of the HyperManual system, the focus of the thesis. Section IV gives an overview of the success of the product. Section V describes

---

\* Smalltalk/V is a regestered trademark of Digitalk, Inc.

some of the problems with hypertext systems and some possible extensions to the HyperManual system. Appendix A is a user manual for the system and appendix B is a hierarchy of relevant classes in the system. Appendix C is a glossary.

## **II. Background**

### **History of Hypertext**

Hypertext is a computer-based environment that provides a single coherent interface to a primarily text database. Hypertext allows non-linear, user-directed access to information. This is fundamentally different from the basically sequential method of presentation of (and access to) information that is provided by most media (including this paper). In books, articles, movies and television, the organization, depth, and sequence of presentation of information is largely controlled by the developers (authors). Hypertext provides an alternative: it allows non-sequential construction of, access to, and interaction with data. In a hypermedia system, the documents referenced are not restricted to purely textual information. [Conklin 1987] [Fiderio 1988]

The original concepts for hypertext were developed by Vannevar Bush in 1945. At that time, he published a description of a data storage structure called "memex" that was modeled on the associative behavior of human memory. Bush described memex as

"a device in which an individual stores his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory." [Smith & Weiss 1988, p. 816]

Bush envisioned a highly interactive, combined text-and-graphics-based system that the user could add to, modify, and restructure according to his current needs. He felt such a system, with free, non-hierarchical links between data objects, would capture some of the associative capacity of the human mind, without the mind's limited capacity for the retention of detail. [Augarten 1984, chs. 3 & 5] [Conklin 1987] [Fiderio 1988] [Smith & Weiss 1988]

The first steps toward implementation of a system such as Bush described did not occur until the 1960's. In 1963, Douglas Engelbart of the Stanford Research Institute published "A Conceptual Framework for the Augmentation of Man's



Intellect." He proposed a highly interactive system in which the user and the computer would be "dynamically changing components in a symbiosis which [would have] the effect of amplifying the native intelligence of the user." [Conklin 1987 p. 22] By 1968, Engelbart had implemented some of his ideas in a system called NLS (oN Line System). NLS was designed as a storage, development, and communications tool for group research. All the details of the development of a project: memos, designs, reports, alternate explanations, criticisms of ideas, specifications, programs, documentation, etc., could be processed, and stored in the same system. [Conklin 1987] [Fiderio 1988]

NLS had typical hypertext features: *non-linear access* to the text in the database and *windows* that provide a visual contextual frame or view of both the structure of the database and the text located in the nodes of the database. NLS also incorporated a new input device developed by Engelbart that has become an indispensable support for graphical interfaces: the mouse. In later incarnations of NLS (now called Augment), Engelbart introduced other features such as a context-dependent help system and multiple windows. [Conklin 1987] [Fiderio 1988]

During the same period, Ted Nelson introduced the concept of an on-line literary environment that would contain all the world's great literature. His system, called Xanadu, is still in the development stage. Nelson has concentrated on designing the "back end" of the system, how documents will be stored and linked. Given the scope of the intended database (text, movies, audio, and graphics) the design and implementation of the back end has been a major task. Recently, Nelson has discussed using *Guide* (described later) as a "front end" to his system. [Conklin 1987] [Fiderio 1988] [Harriman 1987]

## Current Systems

### IBIS

Problem exploration systems are designed to facilitate the exploration of complex problems. They provide a computer-based medium in which all aspects of the exploration of the problem and the search for the solution to the problem can be supported. A typical example of such a system is IBIS (Issue-Based Information System). IBIS supports the interchange of ideas by combining

features of teleconferencing systems with the database manipulation and construction facilities of hypertext. IBIS defines a set of node types and a set of link types to support the development of issues, positions regarding issues (criticisms), and responses to criticisms. The nodes are of three types: issues, positions, and arguments. The type of a link, one of *responds-to*, *questions*, *supports*, *objects-to*, *specializes*, *generalizes*, *refers-to*, and *replaces*, defines the relationship of one node to another. [Begeman & Conklin 1988] [Conklin 1987]

The IBIS problem exploration system differs significantly from the NLS/Augment and Xanadu hypertext systems previously described. IBIS nodes are not designed to contain one idea as NLS/Augment nodes are. In contrast to both NLS/Augment and Xanadu nodes, IBIS nodes are typed to define content, and the links are typed to define relationships among nodes. These features are intended to help direct exploration of the problem and simplify use of the system. Unfortunately, the same features make the link construction process more difficult and time consuming. [Begeman & Conklin 1988] [Conklin 1987] [van Dam 1988]

### KMS

KMS (Knowledge Management System) is a hypertext system that concentrates on providing an integrated, highly standardized, user interface. Nodes in the KMS system are fixed-sized, non-scrollable frames that may contain text, graphics, and/or pictures. Link sources are separated from textual or graphical information and are indicated by the character ° followed by text that describes the link destination. A menu of command options is presented at the bottom of each window. [Akscyn et al. 1988]

Most operations within the system, including frame to frame navigation, and the creation, editing, and linking of frames, are controlled by a mouse. The mouse-controlled cursor is context-sensitive, so that over 90% of the interaction with the system is achieved by a single point-and-click of the mouse. For example, the mouse can be used to highlight a block of text and then move that text by "dragging" it with the mouse to a new location within the frame, or to a location in some other frame, all without the use of auxiliary commands or menus. In this way, the boundary between the browsing, linking, and editing operations has been minimized. [Akscyn et al. 1988]

Another feature of the KMS system is very fast response time. The time required to access and display a frame averages less than 0.25 seconds. A fast response time is required since only one or two frames can occupy the screen at a time. If it took a significant amount of time for a user to access a referenced frame, he would be less likely to do so and the benefits of the system would be lost. [Akscyn et al. 1988]

### NoteCards

NoteCards is a workstation-based, general hypermedia environment. It consists of four basic parts: notecards, links, fileboxes, and a browser. [Halasz 1988]

A *notecard* is a window that can contain information of any form: text, graphics, and/or pictures. It is based on the concept of a 3x5 card, but it may contain any amount of information. The notecards are loosely typed by the form of the data they contain, but, in theory, the type of a link is unrestricted. [Halasz 1988]

*Links* are used to connect nodes together. A link is unidirectional (*from* one node *to* another) and has a user-specified label to indicate the relationship between the two nodes. [Halasz 1988]

*Fileboxes* are special notecards that are used to organize collections of notecards (including other fileboxes) into useful units. Fileboxes provide a method of imposing a hierarchy on a collection of information. [Halasz 1988]

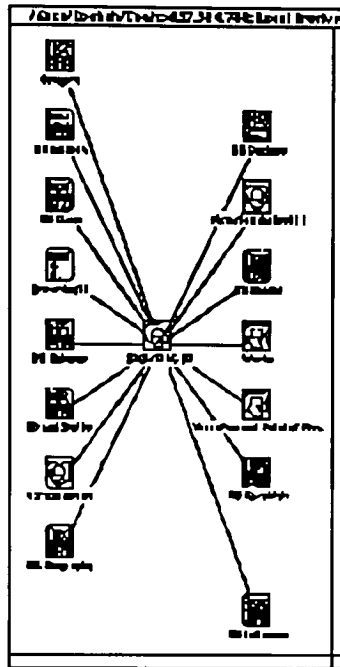
The *browser* is a specialized notecard that gives a graphical view of the structure of the database. Individual notecards are represented as titled boxes and the links (including the link's direction) are represented by arrows that point from one box to another. One navigates through the network by using the mouse to point to a box one wishes to see. Currently active notecards are highlighted in the browser. [Halasz 1988]

## Intermedia

Intermedia is a combined authoring and browsing system with multiple levels of access privileges: read, annotate, write. The environment *looks* the same to all users, i.e., all the menus and menu selectors are there, but the number of active capabilities for a given user are determined by that user's access privileges to a given web and set of document files. Low level users may only have the ability to read a hyperdocument, while high level users can read, write and annotate (the term for creating links in Intermedia) a hyperdocument. Users are further differentiated into those that may only add links and those that may both add and delete links. [Conklin 1987] [Meyrowitz 1986] [Yankelovich et. al. 1988]


In addition to link creation and traversal functions, Intermedia has five document creation applications. *InterText* is a basic word-processing application with functionality similar to Apple's MacWrite. *InterDraw* is a 2-D graphics editor much like Apple's MacDraw. *InterPix* is an application used to display bitmapped images. *InterSpect* is a 3-D viewer that allows users to rotate and zoom 3-D models. *InterVal* is a specialized editor for creating time lines. [Yankelovich et. al. 1988]

To open a hyperdocument in Intermedia, the user first double clicks on the icon of the web he wants to view. If the user has the proper access permission, the web will open an empty *local map* window. The local map graphically keeps track of the current position of the user in the web. The current position in the web is indicated by the icon in the center of the window. The lines radiating from it indicate links from that document to other documents. The icons on the left and right side of the window represent the destination documents. After the local map is opened, the user goes back to the folder that contained the web, and opens one of the associated documents. When the document opens, the local map will update itself in relation to this document. [See Fig.1] The opened document will contain the link icons associated with the web that was opened. [Yankelovich et. al. 1988]



**Fig. 1 -- Intermedia: Local Map**


User interaction with all applications in the Intermedia system is based on subject/verb (postfix) interaction syntax. This style is familiar to users of graphical environments such as Microsoft Windows and Apple's Macintosh operating system. The user selects an object then asks that some operation be performed on that object (such as *copy*). This model is extended to the link manipulation operations to keep the interface consistent. [Meyrowitz 1986] [Yankelovich et. al. 1988]

In Intermedia, a link source may be located anywhere in a document and the destination may be anywhere in the same document or in a different document. Source and destination locations are called *blocks*. The size of a block may vary from an insertion point to a whole document. The existence of a source or destination block is indicated in the document by a block icon  inserted into the document near the block. [Yankelovich et. al. 1988]

Each link, in addition to information about the source and destination documents, has a set of descriptors that further define it. The creation date and user name are automatically entered for link documentation. All links also contain an explainer field that can be used to give a user some information about

where they will end up if they follow a link. The default data for a link explainer is an identification number generated by the system. In a future extension, information such as keywords in the link's associated block will be added to the link. This information would make it possible for a user to search for all links that had a certain set of keywords. [Yankelovich et. al. 1988]

In Intermedia, the set of links (the *web*) for a hyperdocument is not stored within the original documents but is a separate file. Thus, a set of documents may be referenced by any number of different webs, where each web defines a particular view on a set of documents. [Yankelovich et. al. 1988]

Throughout the Intermedia system, a common user interface is maintained to minimize the number of new concepts the user must learn to use the system. The link construction process is uniform across applicable media types. The user selects the source region, chooses the **Start Link** command, selects a destination region, and chooses the appropriate link completion command. Likewise, to follow a link in any type of document, the user clicks on a link icon  and selects the **Follow** command. Alternately, the user can double click on the icon, and the system will follow the link automatically. This simulates the standard Macintosh mechanism of opening an application by double clicking on its icon. [Meyrowitz 1986] [Yankelovich et. al. 1988]

More than one link may emanate from the same source. If a user clicks on a link icon that is a source for more than one link, a dialog box pops up that lists the alternative destinations and their respective explainers. The user can choose one link and follow it or cancel and return (automatically) to the source document. If the icon is instead a source for only one link, the link is automatically followed. [Yankelovich et. al. 1988]

## Guide

Guide was the first commercially available true hypertext system for the PC. It combines browsing and authoring capabilities in a single application. While it provides no image creation facilities, graphics can be inserted into any *Guideline* (a hypertext document created in Guide). Guide provides four types of linking capabilities, basic text editing capabilities, and the ability to import graphics. [Baum 1990] [Owl International 1989] [Harriman 1987]

The hypertext functionality of Guide is supplied by four link types: Expansion, Note, Reference, and Command. The existence and type of a link button in a document is exhibited by the font style (in text) and by changes in the shape of the cursor as the user moves it over a link button (in both images and text):

Type Style	Cursor Shape
<b>Expansion Button</b>	⊕
<u>Note Button</u>	✱
<i>Reference Button</i>	➞
<i>Command button</i>	➞

These clues help the user to know what will happen if he clicks on a button. [Owl International 1989]

The expansion button behaves in a manner similar to the *Outline* utility in Microsoft Word. When the user clicks on an expansion button, the hidden data is expanded. This data may then be closed by clicking on it and the document will return to its former state. Visually, when the button is pressed, a block of text and or graphics is added to the current document after (or replacing) the button. This can cause some disorientation if the expansion takes up more than a small part of the current window as it is not always clear how the new state of the document relates to the state before the expansion. [Baum 1990]

The *Note* button is used to add brief textual or graphical explanations to a Guideline. If the user presses and holds down the mouse button over a note button, a pop-up window appears and remains open as long as the mouse button is held down. This presents a problem if the user wants to make a note about the contents of a *Note* window since he must hold down the mouse button to continue viewing the note. [Owl International 1989]

*Reference* buttons are used to link one Guideline to another or to link within a Guideline. A reference button is used instead of an expansion button when the data referenced is the destination of more than one source, (i.e., there are many

links to the same information). When the user follows a reference *within* a given Guideline, the Guideline scrolls to the appropriate spot and expands *expansion* buttons (if necessary) to show the destination information. If the destination is in another Guideline, that Guideline is opened within a new window, and that window is scrolled and expanded as necessary to bring the referenced information into view. [Baum 1990] [Owl International 1989]

*Command* buttons are used to open other applications from within a Guideline. They may also be used to control any device connected to the serial port. The Guide system includes a simple interpreter to read command scripts referenced by command buttons. [Owl International 1989] [Harriman 1987]

In addition to the browsing capabilities provided by the various buttons, Guide provides two aids that help the user retrace his path within a document. The **Top Level** button on the **Display** menu returns the user to the starting state of the document he currently is in. It does not necessarily return the user to the starting state of the Guideline that he originally started browsing. Each window also has a **Backup** button that to some degree makes it possible for the user to retrace his path, step-by-step, through a document. [Owl International 1989]

Creation of expansions is a relatively straight-forward task. The user first selects the text or graphical object that is to be the source, then chooses **Button** from the **Make** menu. This creates the expansion button itself, and if the source object is text, that text will be bolded. Next the user creates (or imports) the desired expansion in(to) the Guideline, and select **Replacement** from the **Make** menu. This completes the expansion creation process. [Baum 1990] [Owl International 1989]

Note, Reference, and Command buttons are designed to be created in a manner different from expansion buttons. First the user creates the destination object and link, then he creates the source button and link it to the previously created destination object. For example, if the user wishes to create a reference button to link to an image, first he creates a new Guideline document and paste the desired image into the document. Next the user selects the image and chooses **Reference Point** from the **Make** menu. This creates the destination. To create the reference button, the user selects the source object and chooses **Reference**



from the **Make** menu. If the user wants to create another reference to the same destination, the process is different. First the user selects the source of the reference and choose **Reference** from the **Make** menu. Then the user selects **Start Link** from the **Make** menu to indicate that this is the start of a link that as yet has no destination defined. Next the user selects the destination object and chooses **Link Reference** from the **Make** menu. [Owl International 1989]

The Guide hypertext system has a number of limitations and ambiguities.

- 1) The linking information is built into a document, not superimposed on it. This means that a document that has source buttons in it can only be part of one web. If it is used as a part of two webs, those two webs will become tangled into a single web.
- 2) A given object can be the *source* of only one link.
- 3) A graphical object can be the source of only one out-going link. This means that the user can't easily add notes to describe each part of an image.
- 4) There is no explicit way to remove a link. All the user can do is delete the source of the link. Unfortunately, this change may not be conveyed to the destination of the Reference, Note, and Command links. They still act (as indicated by the active menu selectors) as if they are the destination of a link and this can make the link creation process confusing. This situation is the result of having link information stored in the document itself.
- 5) It is not possible to import formatted text into a Guide document and maintain the formatting. Guide only allows its own internal forms of formatting. [Owl International 1989] [Harriman 1987]

### **HyperCard, Toolbook, Plus, etc.**

There is a class of applications (HyperCard, Toolbook, Plus, etc.) in which it is possible to create simple hyperdocuments. These programs are not, however, designed specifically for the creation of hyperdocuments. They provide only a minimal set of hypertext tools i.e., the ability to place a button on a "card" and to link from that button to another "card". To create a hypertext system in one of these applications, the user would need to write his own support tools as scripts. Scripts are subprograms in that applications built-in language.

Because of the programming requirement and the lack of hypertext support tools, these applications are not true hypertext systems.

### Summary

All the hypertext systems discussed with the exception of Intermedia embed the link structure within the base documents. This ties a linked document to a particular web. KMS limits its flexibility by allowing only one fixed size, non-scrollable window on the screen at a time. This forces all linked documents to be small. NoteCard suffers because of its tree structured navigation system. Even though individual notecards are linked, to move from card to card, the user must use the browser card. Intermedia provides a similar supplemental navigation aid with its global and local navigation maps. Unfortunately, these navigation systems become useless in a hyperdocument with a large number of links. There can easily be so many links that it is very difficult to know what documents are linked with a given document. Non of the systems described gives the user control over whether or not link indicators are showing. The user cannot view the document as it stands and when he wishes, see what links exist. It is like trying to read a book that was highlighted by someone else. Those marks interfere with the flow of reading, emphasizing portions of the document that the reader might not normally stress.

## **III. HyperManual Implementation**

### **Design Issues**

There are a number of issues that must be addressed in the design of a hypertext system. These issues fall into two categories: structural and interface.

#### **Structural Issues**

A fundamental issue in the design of a hypertext system is the relationship of the links to the documents. The most common approach is to embed the links within the structure of the linked document. This approach attaches a document to a particular web. It also alters the original document since, link source must be shown by changing the formatting of the text, or inserting some form of link indicator into the document such as a button or an icon.

The alternative is to superimpose the links on the linked documents. Independent links make it possible for there to be multiple views (link webs) on the same document. They also leave the base document unaltered so the fact that the document is part of a hyperdocument is only incidental.

A second structural issue is the all inclusiveness of the hyperdocument construction environment. At one extreme are systems (like Intermedia) that provide complete document creation support in addition to hypertext facilities. The HyperManual system (described later) takes a more limited approach. The HyperManual system concentrates on providing the tools required for the use and construction of a hyperdocument. The base documents are created in standard applications such as MS Word and SuperPaint and imported as RTF and PICT documents.

#### **Interface Issues**

Among the interface issues to be addressed in the design of a hypertext system are:

- 1) the manner in which links are to be made evident to the user,
- 2) the relationship between the link creation and browsing interfaces, and
- 3) the options available to the user for navigating within a hyperdocument.

The approaches taken on each of these issues has a profound affect on the user acceptance and usefulness of the resultant hypertext system.

## **HyperManual System Description**

HyperManual is a two mode hypermedia environment that combines hyperdocument browsing and linking capabilities in a single integrated system. The system is a hyperdocument browsing and creation environment. (Intermedia and to a lesser extent Guide are document creation environments in addition to being hyperdocument browsing and linking environments.) Documents used by the system are created in standard applications such as Microsoft Word or SuperPaint. HyperManual currently imports text in Rich Text Format (RTF) and images in PICT format.

## **Hyperdocument Structure**

As in Intermedia, the web of links for a hyperdocument is separate from the documents themselves. The system superimposes the web on the documents to create the hyperdocument. This makes it possible for multiple views of the same document to exist without giving rise to the problem of separate hyperdocuments becoming joined at the document level (as can happen in Guide).

## **Browsing Mode**

The browsing mode of HyperManual is used for viewing the contents of a hyperdocument. In browsing mode, the two initial methods of access are through the Table of Contents and the Index windows. The Table of Contents simulates a standard table of contents and gives the user access to images and text in the hyperdocument. The Index expands upon the functionality a standard index and gives the user an indexed view of the contents of the hyperdocument. Once the user is viewing a particular document, he has access to related information through links that are superimposed on the document. In the browsing mode, the user can also add his own annotations to a document. A third method of access to a hyperdocument (in browsing mode) is through the AnnotationBrowser. This browser provides a user with an overall view of the annotations he has added to a document.

## **Linking Mode**

The linking mode of the system is used for hyperdocument creation. This mode provides the user with an integrated environment for linking documents. To minimize the cognitive overhead of the linking process, the number of steps in the link creation process and the required user input per step is kept to a minimum.

The basic link creation process involves selecting the link source, selecting the link destination, and saving the link. For links to text, the context of the link destination must also be defined. The system extracts default values for the link explainers from the selected text in the case of a text link or the document name in the case of a link to an image. The linking operations may be performed in any order and may be intermixed with non-linking operations such as opening a file. The user may not, however, save a link until he has performed all the operations necessary to define the link.

## **Uniform User Interface**

To enhance ease of use, there is an emphasis on the uniformity of the user interface. To follow a link in any document, the user clicks in the source selection area and choose the link he wants to follow. All links to documents are created in the same way, the user selects the source, links to the destination, and saves the link. Annotations are added to and revised in all documents the same way. All document windows can be scrolled and resized and may contain any size document (up to the memory limit of the machine).

## **Minimization of System Complexity**

A number of steps have been taken to reduce the overall complexity of the environment. In browse mode, the user can set the maximum number of document windows he wants open at a given time. When a new document window is opened, if the maximum has been reached, the system will close the oldest unmarked window to keep the total number of open windows from exceeding the maximum preferred. The user can protect a window from being automatically closed by marking it. To minimize user disorientation, windows

are automatically stacked when they open. This helps the user maintain a visual ordering of his progress through a hyperdocument.

Link sources in images and text are shown by highlighting the selection area of the link source. The intrusiveness of link sources is diminished by giving the user control over whether they are showing. The user can inquire about the information available at the destination of a link without committing herself to following the link. Clicking on a link source brings up a pop-up menu that contains a list of the link destinations associated with that link source. Each entry in the list defines the type of the destination: Text, Image, or Note. Following the type of each link entry is a short explainer gives the user some idea of what information is located at the destination of the link. [See Fig. 11]

In Link mode, the complexity of the linking process is mostly hidden from the user. Although there are currently six address types and twelve address combinations, link type management is the concern of the system, not the user. The system's contextual link address typing only becomes apparent to the user in the pop-up destination link window and the Index. In both windows, the link type is used to give the user information about the destination document.

An unusual feature of the linking process is the ability to define the context of the information in the destination of a text link. When a link is followed to a textual document, only the portion of the text selected by the link creator is initially shown in the document pane. This helps the link creator to minimize the amount of extraneous information presented to a browsing user. If the user wants to browse the text surrounding the defined context, he can **PageUp** or **PageDown** to see nearby text.

The system gives the link editor the ability explicitly to remove any link in a hyperdocument (unlike Guide). Another feature is the ability to change quickly into (and back from) a *pseudo* browsing mode while in the link mode. This makes it possible for the link creator to test the links he has created exactly as they would appear to a browsing user.

## Annotations

Two innovations in the system are annotations and the Annotation Browser. Annotations are a mechanism that allow a user in the browse mode to add his own notes to a document. The Annotation Browser makes it possible for the same user to view all the annotations he created in a hyperdocument in the context of the place in the document each annotation was inserted. The user can also both add new annotations to, and remove existing annotations from, a hyperdocument. Using annotations in combination with the Annotation Browser, a browsing user can create his own personal view on a hyperdocument.

## Class Hierarchy

The following is a description of the classes in the HyperManual system and the functionality provided by instances of each class. (See Appendix A for a detailed description of how to use each element of the system and Appendix B for the class hierarchy for the HyperManual system.) The classes are divided into three categories by function: Dispatchers, Models, and Panes. This partition is the result of the Model, Pane, Dispatcher (MPD) interaction that is the basis of window interaction in the current version of Smalltalk/V Mac. (Smalltalk/V PM and Smalltalk/V Windows use a simpler Application - Pane model where the functions of the Model and Dispatcher are combined into the Application.) In the MPD model, a Pane provides a view of the information in a Model, a Dispatcher handles the input to a Model, and a Model determines what the effects of changes to the data in the model mean. [LaLonde 1989] [LaLonde 1991]

The Model category is further divided into Window, Pane Support, and HyperMedia models. Window models are closely tied to particular windows (and, in the case of HyperImage and HyperText, panes and windows). Pane Support models provide functionality to the Panes while HyperMedia models define the functionality of the hypermedia system itself.

## Dispatchers

The dispatcher classes in the system redefine a few of the input processing functions of their super classes to make those functions appropriate for their

associated panes. This mostly involves responding appropriately to inquiries as to whether the contents of the associated pane have been modified and making sure the contents of the pane scroll properly.

### **Panes**

#### **HyperTopPane**

One of the features of the system is that it can control the number of windows open at a time by automatically closing the oldest window. The user can keep a window from being automatically closed by the system by locking that window. The ability to lock (and unlock) a window is provided by the HyperTopPane class.

#### **ImagePane and ImageLinkPane**

The ImagePane and ImageLinkPane add to the GraphPane class the ability to highlight a selected area of the pane. The ImagePane is used by the system to show images in the browsing mode. The ImageLinkPane adds to ImagePane the ability to show the linking process.

#### **HyperTextPane and HyperLinkPane**


The HyperTextPane is used to show formatted text when the system is in browsing mode. The HyperLinkPane is used to show the linking process in formatted text. The ImagePane, ImageLinkPane, HyperTextPane and HyperLinkPane all provide support for the ability to show multiple selected areas in a single pane.

### **Models**

#### **Pane Support Models**

The Pane Support Models provide specialized functionality for the image and text panes in the system. The HyperStringModel serves as the textholder for the formatted text in the HyperTextPane and HyperLinkPane classes. The Selections classes are used to provide visual feedback as to the existence of a link in an image or text pane. They are able to detect *noteMarks* as special selections and



display them as  to differentiate them visually from standard link sources which are simply highlighted regions in a document. The MultiSelections classes are used to display multiple selections in a single pane.

### HyperMedia Models

The HyperMedia models provide the hypermedia functionality in the system. Each model defines or controls some aspect of the web of a hyperdocument.

The Address classes define the characteristics of the source, destination, and context of the link. An address contains the name of the file referenced, the selection rectangle, and for the source and destination, an explainer. The abstract superclass HyperAddress contains the methods that access this data. The primary function of the HyperAddress subclasses is to distinguish the type of an address which in turn determines how the information contained in an address is used.

The Link class defines the structure of the links used to 'glue' a hyperdocument together. Each link contains the address of the source, destination, and context (for text) of the link.

The Path and WindowStruct classes provide the data structures and associated operations necessary for the retrace facility in the system. The *retrace* facility makes it possible for the user to retrace the sequence of links he followed to get to his current location. Each instance of a WindowStruct holds the information necessary to return a window to a given state (including reopening the window if necessary). A Path holds the sequence of steps, in the form of a list of WindowStructs, that a user takes as he follows links from window to window.

The primary task of the Manual class is managing the linkTable. (The name of the class is a result of the initial purpose of the system which was to create a hypertext version of a technical manual.) The linkTable is a dictionary of link dictionaries and collections indexed by file name. The linkTable is the glue that binds together a set of documents into a hyperdocument. When a link is created, the manual makes sure the link information is valid before it saves the link in the linkTable. The manual also handles filing in and filing out the linkTable. In

addition, the manual handles certain system support functions such as the retrace facility, automatic window closing, and preference setting operations.

### Window Model Classes

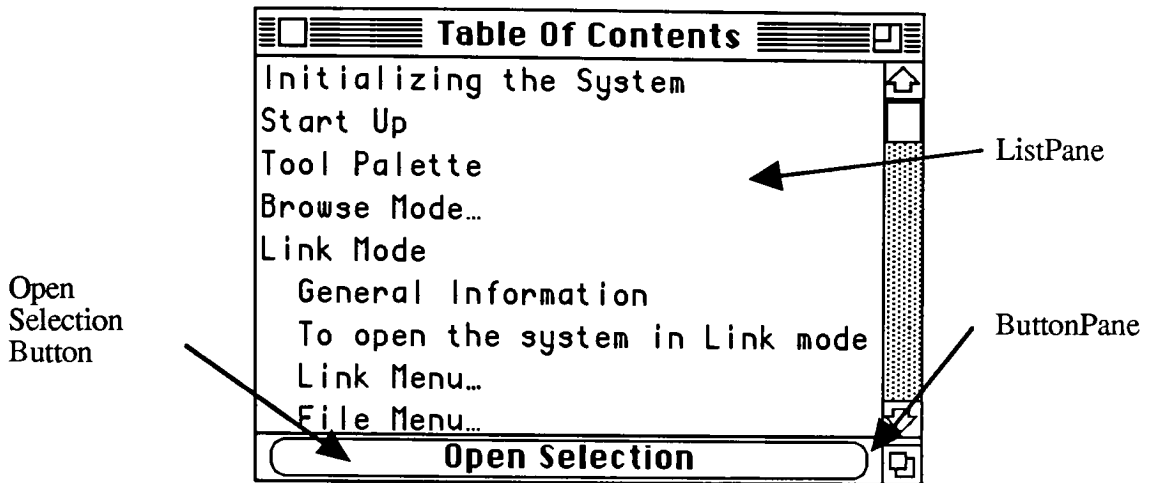
Window model classes implement the actual views on the system. The simplest class, `HyperFormViewer`, supports the `HyperImage` class by defining the parameters necessary to open a window that contains an image pane. There are eight window models in the system: `Annotation`, `ManualIntroPane`, `HyperImage`, `HyperText`, `TableOfContents`, `Index`, and `AnnotationBrowser`.

Class `Annotation` implements the two forms of annotation window: the annotation creation window and the annotation revision and removal window. This class also creates a new link (if necessary) and saves the new or revised information in the link and asks the manual to save the link.

Class `ManualIntroPane` implements the first layer of interaction with the system. It creates a tool palette that contains subtitled icons that allow the user to activate the various browsing and linking facilities the HyperManual system provides.

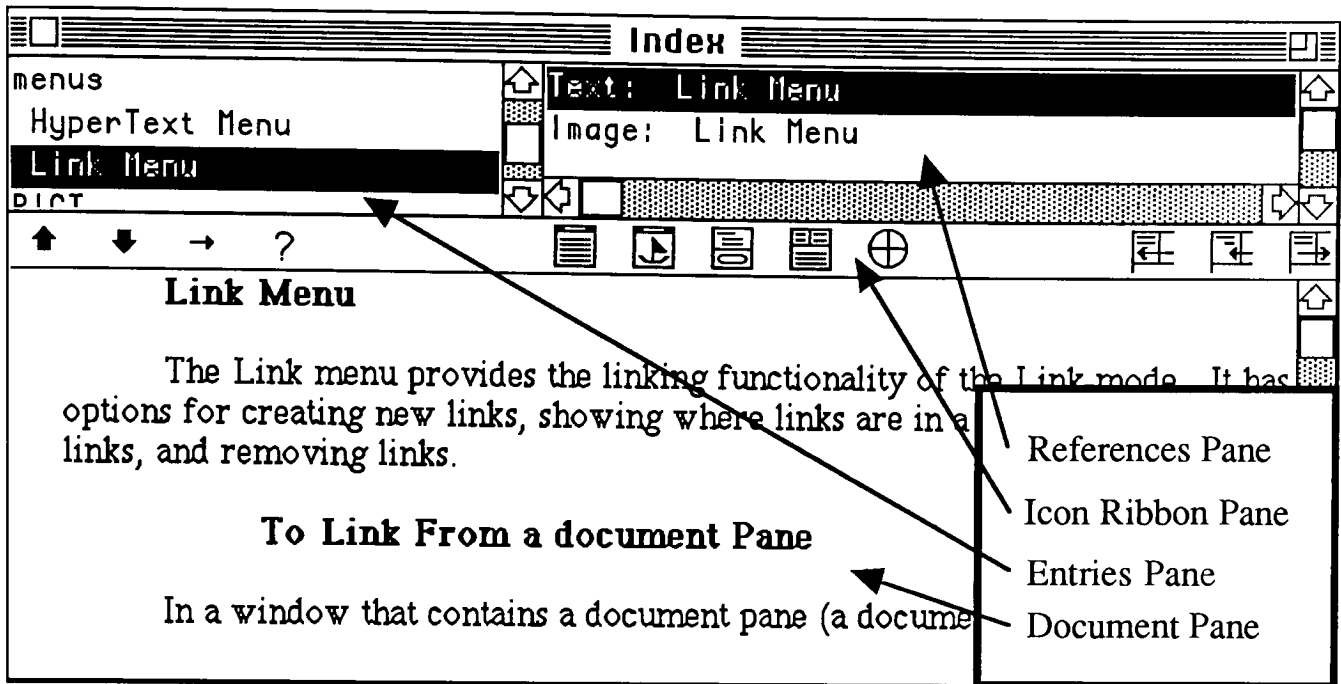
The `HyperImage` and `HyperText` classes implement the primary interface between the user and the actual information in the system. Class `HyperImage` controls the link creation process for images, the proper processing of mouse clicks in the dependent pane, and the highlighting of the selections in an image. The `HyperText` class controls the same operations for text.

The `TableOfContents`, `Index`, and `AnnotationBrowser` classes furnish the system with various access methods to the information data base of a hyperdocument. The `TableOfContents` class mimics the functionality of the table of contents in a book. It provides access to the images, figures, and text in a hyperdocument. The `TableOfContents` class also controls the link creation process for the links that emanate from the table of contents. The `TableOfContents` window contains two panes: a `ListPane` that contains a list of the directly accessible portions of the hyperdocuments, and a button pane that contains the **OpenSelection** button which causes a selected section of the hyperdocument to be opened. [See Fig. 2]



**Fig. 2 -- Table of Contents**

Class Index implements a window on the contents of a hyperdocument that expands upon the functionality of an index in a book. The index window is composed of four panes. The first pane (the entriesPane) is a ListPane that contains entries for each indexed element in the hyperdocument. Each entry may have any number of subentries (and each subentry may have subentries). The second pane (the referencesPane) is also a ListPane and contains the references associated with the selected entry in the entriesPane. The third pane (the icon ribbon) is an IconButtonPane that contains icons for quick access to various operations relevant to the index. The fourth pane (the document pane) contains the text or image referenced by the selected entry in the referencesPane. The Index class also controls the index list creation process and the linking process for links within the index. [See Fig. 3]



**Fig. 3 -- Index (in Link Mode)**

The AnnotationBrowser class provides a view on the contents of the hyperdocument that simulates the action of putting PostIt notes in a book. The AnnotationBrowser has seven panes. The first pane contains an entry for each document in the hyperdocument. The second pane contains a button that allows the user to open the document selected in pane one. The third pane contains a list of the annotations added to the document selected in pane one. The fourth pane contains the text of the annotation selected in pane three. The fifth pane contains buttons for saving or removing the selected annotation. The sixth pane (an iconRibbon) contains the icons for functions relevant to pane six. The seventh pane contains the text or image that 'contains' the annotation selected in pane three. In the AnnotationBrowser window, the user can view existing annotations in a hyperdocument. The user can also use the AnnotationBrowser window for adding new annotations to the hyperdocument and removing existing annotations from the document. [See Fig. 4]

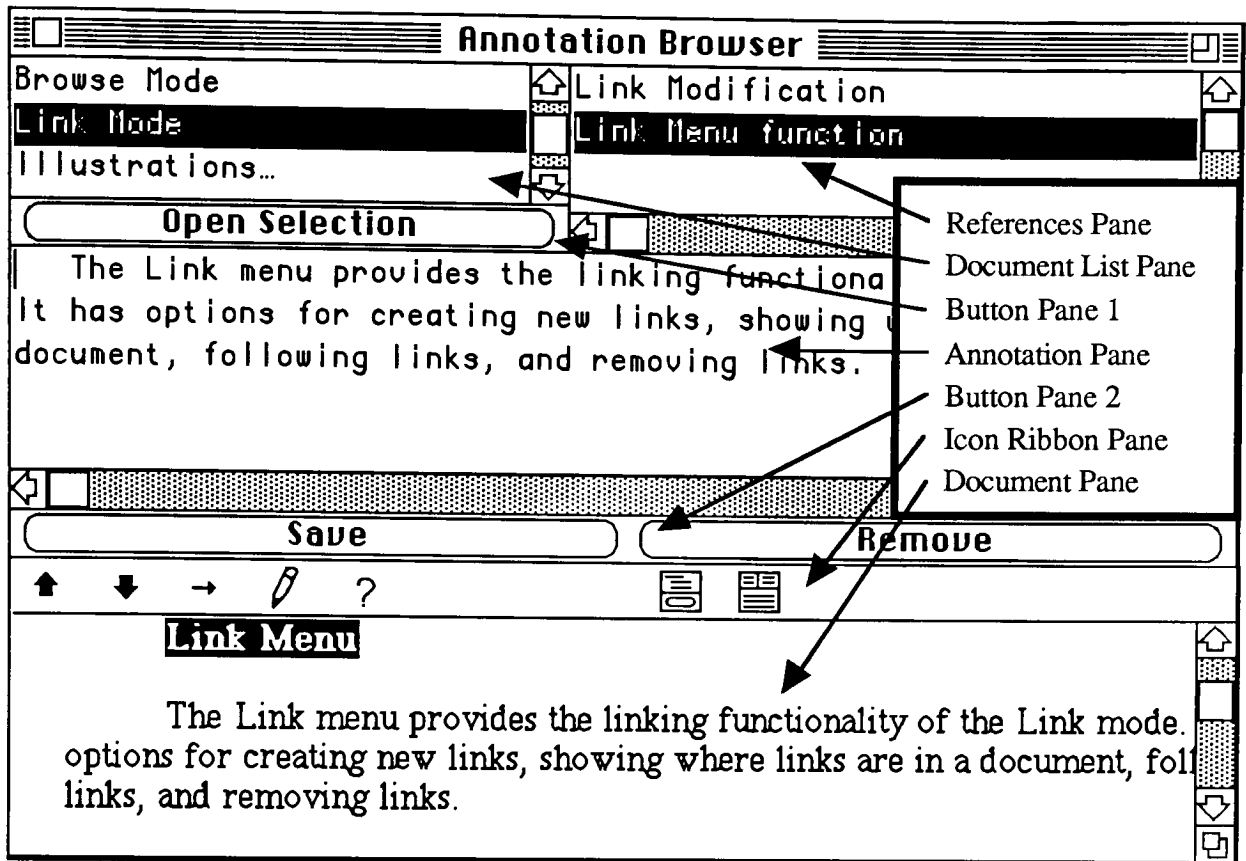


Fig. 4 -- Annotation Browser

## Chronology

The original ideas for the design of the HyperManual system were, in part, a combination of the design ideas in KMS, Intermedia, and the general layout of a text book. Like KMS, the system would have a list of link sources for each 'page' of the document. The system would have some form of document similar to Intermedia's local map to give the user a visual clue about where he is in the system. The system would also include the standard quick access methods of a text book, i.e., a table of contents and an index.

One of the design considerations was that the system had to be able to work with pre-existing documents that had not been written as hypertext documents. It was important that the amount of work necessary to convert a document to a hyperdocument be minimized. As a result, we abandoned the simplicity of the KMS design for a more complex superimposition of the links on the original

document as in Intermedia. The approach of Guide where the links are embedded in the text was too limiting and would have made the document creation process more time consuming. We also abandoned the concept of a document map since, as soon there were more than a few links exiting a document, a map of the links became hopelessly cluttered.

An early question was 'How should the location of link sources be shown in a document?' The first idea was to change the formatting of the document as Guide does to indicate link sources. There are a number of problems with this approach. First, we had decided to maintain as much of the documents original formatting as possible. In all likelihood, special formatting for link sources would get confused with the pre-existing formatting of the text. Second, it would require modification of the original document. This would tie a document to a particular web of links. Third, special formatting of link sources would alter the visual flow of the original document; it would be like trying to read a book that someone else has highlighted. Fourth, since links in images would need to be depicted in a different manner, it would introduce an unnecessary inconsistency into the user interface.

Instead, we chose to display link sources as highlighted regions in a document. This made it possible to keep the links completely separate from the linked document. Since the user can turn the display of the link sources on and off as desired (unlike Intermedia), the user has an unimpeded view of the original document.

Other issues, such as the design and functionality of the Index and Annotation Browser, have evolved over time. For instance, the document pane of the Index originally could only display text. It became clear that the Index would be much more useful if it could display both images and text. The ability to make such progressive changes was facilitated by the Smalltalk/V object-oriented programming environment.

Midway through the project, the basic hierarchical structure of the system was redesigned to reflect better the functional divisions in the system. Originally, the class Manual was the model for many of the windows in the system; only the Index and Table of Contents windows had their own models. This made the Manual class unwieldy. To simplify Manual class, the Annotation, HyperImage,

and HyperText classes were created, each as a model for its associated pane. This degree of restructuring would have been extremely difficult in a language that was neither class-based, nor hierarchical.

## **IV. Results**

### **Integration**

The HyperManual system depends on a large number of other systems (groups of related classes within Smalltalk/V) for numerous support functions. In the future, more support systems will be added and HyperManual may be integrated with other applications such as Microsoft Word and Excel.

### **Modifiability**

During the design and implementation process, HyperManual has undergone some significant modifications. One of the basic requirements was that the system modular and extensible. Although it is likely that some of the anticipated extensions will cause significant changes to some classes and subsystems (e.g., the change from a stand alone to a networked system will change parts of some of the link management methods) the effect of those changes should be localized. It is also expected that some of the classes and subsystems developed for this product will be used as parts of other projects (e.g., the MultiSelection class and its subclasses and the MultiSelectionListPane).

### **Criticism of Final Product**

The user interface design and implementation presented the most difficult problems in the system. It is also the source of the most persistent ongoing problems. For example, a continuing problem is maintaining the current state of the links when they are showing in a document. Some pop-up windows and dialogs do not inform other windows of their birth and demise. When this happens, a highlighted area that was partially covered by a pop-up may not redisplay itself correctly. This is particularly a problem with document panes containing images. Another shortcoming with the functionality of the system is that the user can't paste text into Smalltalk/V dialog boxes. This creates an interface inconsistency since many of HyperManual's dialogs are pastable Macintosh dialog boxes.



To encourage the user to follow links, the action of following a link must happen very quickly. Because the process of reading in formatted text from a file is so time consuming, the text for a hyperdocument is stored in the Smalltalk/V image. Storing text in the image takes up a great deal of RAM; therefore, the maximum size (in text) of the hyperdocument is limited.

In Intermedia, Guide, and NoteCards, the linking, document creation, and revision operations can be combined (at least for text). HyperManual assumes that all the document files it uses are in a frozen state. If the user modifies a document that already has links in it, it is very likely that he will need to redo the links to and from that document. The only modifiable information (aside from the links) that does not cause this problem is data contained in the pop-up notes a linker creates and in the annotations a browser creates.

## Product Overview

The system as a whole functions very well. For the most part, actions involved with following a link take place in a reasonable amount of time. The only case where link traversal ordinarily slows down is when a large image is to be displayed. This action may take a second or two.

A problem does exist in the processing of the initial request to show links if there are a large number of link sources (greater than 100) in a document. It takes a significant amount of time (more than a second) for the link selection regions to be calculated. As a result, the user may feel that the system did not get the *Show Links* request he sent. If the user then sends another *Show Links* request (say by clicking on the show / hide link icon) the system interprets that as a request to hide the source selection areas. Consequently, the link sources will not be shown and the user will wonder what happened.

There are a number of partial solutions to this problem. The first is to speed up the link source selection generation algorithm, and some steps have been taken in this direction. The second is to give the user more feedback as to the current state (show links or hide links) and more information about the amount of time the process is taking so he knows he should wait. The third approach is to decrease the size of the individual (text) documents. This would limit the average

number of links a document would usually have so that the time necessary to complete the link source selection generation process would not be a problem.

The Index is an unanticipated success. In browse mode it is very easy to scan quickly through a large number of references and find the needed information. This is largely a result of having the document pane be part of the Index window. If a particular reference is not the one the user wants, he just clicks on the next likely reference and it is immediately placed in the document pane. There is no need to close down a window and go back to the Index to look for a reference. It is also useful that the document pane displays images as well as text. For example, the user can quickly see how a component fits into a system and then go to a textual description of the component. Since the document pane in the Index functions just like (actually is just like) any other document pane in the system, the user can not only scan indexed information; he can also follow links out of the index to independent document windows.

In the Link mode, the Index can simplify the general linking process. The user can index (perhaps temporarily) a group of documents (or document sections) that he wants to link and use the Index as a quick means to switch between the various documents. This simplifies the linking process since the user only needs one window open to link among the set of documents. This is often easier and faster than having a number of separate document windows open and switching between them.

Another success is the Annotation Browser. It gives the user in browse mode the ability to create his own personalized view on a hyperdocument. Annotations, in general, also make it possible for the user to insert his own information into a hyperdocument without worrying about affecting the original document. This will become especially important when the system becomes network based.

## **V. Conclusions**

### **The Hypertext Concept**

A hypertext system can have a number of advantages over other data management systems (computer-based or not). Tracing references in a hypertext system is far easier than trying to locate physical copies of material, no matter how extensive the library. Within a hypertext system, it is relatively easy to construct different versions of the same document simply by creating different paths through the data. [Conklin 1987] [Frisse 1988] [Fiderio 1988]

Although a hypertext system provides tools that make it easier to manage and manipulate data, it is still a technology with unresolved problems and limitations. One problem with current hypertext systems is the difficulty of finding one's way through a large network of densely connected documents. In each document there can be so many choices, it may be difficult to choose which path to follow. One can quickly become lost in peripheral issues. The link structure of a hyperdocument can offer a person freedom where he may need guidance. [Conklin 1987] [Fiderio 1988]

Hypertext systems also face a problem common to all database management systems: keeping the database current. The interconnections between bits of information make it difficult to add or delete significant quantities of information without overhauling the database. Documents in a network may also become overloaded with obscure references and connections that destroy the effectiveness of the system. [Begeman & Conklin 1988] [Fiderio 1988]

### **The HyperManual System**

The HyperManual system succeeds in its basic goal of simplifying the act of locating needed information in a complex technical document. The system can provide quick, flexible access to information through multiple views on or access methods into a hyperdocument. Through Annotations and the Annotation Browser, the browsing user can add his own information to a hyperdocument and create his own view on its contents. If a hyperdocument is properly linked, a user should have little trouble finding the information he needs.

One characteristic that sets the HyperManual system apart from other hypertext systems is the emphasis on providing the user with multiple access avenues to the information in the hyperdocument. This increases the likelihood that the user will be able to find the information he wants. The Table of Contents and the Index are tools that are conceptually familiar to the user but they can work much faster and offer more functionality than their printed counterparts.

When all the connections in a hyperdocument are directly from document to document (as in Intermedia and Guide, for example), the function of the links is heavily overloaded. The interdocument links must 1) provide the paths through the hyperdocument, 2) link to related issues, and 3) link to definitions and descriptions. The inclusion of the Table of Contents and Index (and, in the future, the Text Browser) in the system means that the interdocument links can be more focused and less numerous. Also, the linking process can be divided into logical subtasks, (i.e., definition linking from the Index) simplifying the link creation process.

No matter what tools are available for the construction process, the creation of a complex hyperdocument will be a difficult process. We need to develop and evaluate linking models to discover how a hyperdocument should be constructed so it will be a useful information acquisition aid that will be used by choice.

## **Future Extensions**

There are a number of possible extensions to the current system. Many are primarily intended to simplify the link creation process. The extensions are discussed in approximate order of current need.

The Text Browser would be a text retrieval system that would make it possible for the user to browse multiple unstructured files concurrently using the 'key word in context' approach to guide the search process. The system should incorporate filters that help the user to prune intelligently the potential subject categories to focus their search. This tool could be used either by itself or in conjunction with HyperManual as an alternate information access method. If the TextBrowser system uses HyperManual's HyperTextPane to display the text, the

user could use the Text Browser to locate an item of interest within a document of a hyperdocument and follow available links out of that document.

The Index Builder would combine the Text Browser and the Index into a single integrated window. The user would use the text browsing functions to generate a candidate index list. Words from this list could be added to the index and linked to the references the Text Browser portion found.

The Link Editor would be a five pane window with two list panes, two document panes, and an icon ribbon pane. The first list pane would contain the link sources in the selected file. The second list pane would contain the destinations for the link selected in list pane one. Document pane one would contain the document that contains the link sources contained in list pane one. Document pane two would contain the destination document for the link destination selected in list pane two. This window could be used to scan quickly the links in a file and to add to, change, and remove links.

The Multi-link Creation Automater is a feature that would be available in the Link mode in all windows with document panes. The user would use this to copy a group of links from a given source and 'paste' those links to another source.

The Presentation Manager system would be a way for the hyperdocument author to control how one or more windows are displayed on the screen and to control external devices related to a particular hyperdocument. The first step would be adding the ability to specify the size and location of where a window appears on the screen. The next step might be to write a mini interpreter that could read a script. This script might be used to control external devices or to set up linked windows where actions in one window would affect the state of dependent windows.

Annotations are currently stored in the link table of a hyperdocument. This is appropriate only for a single user stand alone system. On a multi-user system or a networked system, each user should be able to have his own personal set of annotations. The separation of annotations from the regular links would be the first step towards developing a networked system.

For wider potential distribution, the HyperManual system should be converted to Smalltalk/V for Windows. This involves: 1) merging the model and

associated dispatcher classes and making them subclasses of ViewManager, 2) changing the open methods for windows, 3) changing any Macintosh specific classes and methods to Windows specific implementations, and 4) in general, making it work on the new platform.

The first step towards networking the system to separate annotations from other links. Next, the document and link access methods must be modified to handle interactions with a server. After that, the standard file problems of who owns, who may modify, and who may access a hyperdocument must be solved.

It may be useful to know how often a link is traversed. Data about link usage might be used to help develop models that could be used to help design the link structure of hyperdocuments.

It would be possible to construct a word processor in Smalltalk and incorporate link management capabilities into that system. It would, however, be very difficult to make it fast and flexible. A better approach may be to use a currently available word processor such as Microsoft Word and find a way to read and write files that embed link information as hidden text in the document. The document could be filed out, edited, and then read back into the system. When the system read the document back in, it would adjust the links as necessary and flag the links that had been modified by the linking process.

For certain hyperdocuments, such as technical manuals, it might be useful to offer views of the information in the hyperdocument at various levels of detail. An expert might only want links related to concepts while a novice would need links to tutorial explanations. The user, while he is in the hyperdocument, should be able to change the viewing level the system provides to conform to the extent of his knowledge in a particular area.

## HyperManual Bibliography

- Akscyn, Robert, Donald McCracken, and Elise Yoder. "KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations." In *Communications of the ACM*, July, 1988, pp. 820-835.
- Augarten, Stan. *Bit by Bit*. Ticknor & Fields, 1984.
- Begeman, Michael, and Jeff Conklin. "The Right Tool for the Job." In *Byte*, October, 1988, pp. 255-266.
- Baum, Nancy. "Software Reviews: Hypertext Program." In *Library Software Review*, Jan. - Feb. 1990, pp. 45-49.
- Conklin, Jeff. "Hypertext: An Introduction and Survey." In *IEEE Computer*, September, 1987, pp. 17-41.
- Fiderio, Janet. "A Grand Vision." In *Byte*, October, 1988, pp. 237-244.
- Frisse, Mark. "Searching for Information in a Hypertext Medical Handbook." In *Communications of the ACM*, July, 1988, pp. 880-886.
- Halasz, Frank. "Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems." In *Communications of the ACM*, July, 1988, pp. 836-852.
- Harriman, Cynthia. "Hypertext Comes to the Small Screen." In *The Seybold Outlook on Professional Computing*, March, 1987, pp. 14-18.
- LaLonde, Wilf and John Pugh. "Pluggable Tiling Windows." *JOOP*, September/October, 1989. pp. 57-66.
- LaLonde, Wilf and John Pugh. "Windows, panes, and events in Smalltalk/V PM." In *Journal of Object-Oriented Programming*, February, 1991, pp. 57-66.
- Meyrowitz, Norman. "Intermedia: The Architecture and Construction of an Object-Oriented Hypermedia System and Applications Framework." In *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications OOPSLA86*, ACM, September, 1986, pp. 186-201.
- Owl International. *Guide* Owl International, Inc. 1989.
- Smith, John, and Stephen Weiss. "An Overview of Hypertext." In *Communications of the ACM*, July, 1988, pp. 816-819.

## HyperManual Bibliography

van Dam, Andries. "Hypertext 87: Keynote Address." In *Communications of the ACM*, July, 1988, pp. 887-895.

Yankelovich, Nicole, Karen Smith, Nancy Garret, and Norman Meyrowitz.  
"Issues in Designing a Hypermedia Document System." In *Interactive  
Multimedia*, Sueann Ambron and Kristina Hooper ed. Microsoft Press, 1988,  
pp. 33-85.



# HyperManual

## Appendix A -- User Manual

### Table of Contents

Appendix A. User Manual .....	37
Initializing the System .....	37
Start Up .....	37
Tool Palette .....	38
Browse Mode .....	38
Table of Contents .....	39
To Open .....	39
Description and Usage .....	39
Choosing a Heading .....	40
Opening a Document Window .....	40
Index .....	40
To Open .....	40
Description and Usage .....	40
Entries Pane .....	41
References Pane .....	42
Icon Ribbon Pane .....	42
Document Pane .....	42
Annotation Browser .....	43
To Open .....	43
Description and Usage .....	43
Document List Pane .....	44
Button Pane 1 .....	44
References Pane .....	44
Annotation Pane .....	44
Button Pane 2 .....	45
Icon Ribbon Pane .....	45
Document Pane .....	45
Document Windows .....	45
To Open .....	46
Description and Usage .....	46
Icon Ribbon Pane .....	46
Document Pane .....	46
To Follow a Link .....	47
Destination List Window .....	47
Note Window .....	47
Description and Usage .....	47
Annotation Window .....	48
Text Pane .....	48
Button Pane .....	49
To View an Annotation .....	49
To Modify an Annotation .....	49
To Remove an Annotation .....	50
HyperText Menu and Icon Ribbon .....	50
Actions .....	51
To Open or Activate the Table of Contents .....	51
To Open or Activate the Index .....	52
To Open or Activate Help .....	52
To Show (Hide) Link Source 'Buttons' in a Document .....	52
To Lock (Unlock) a Window .....	52
To Page Up .....	52
To Page Down .....	53

# HyperManual

## Appendix A -- User Manual

### Table of Contents

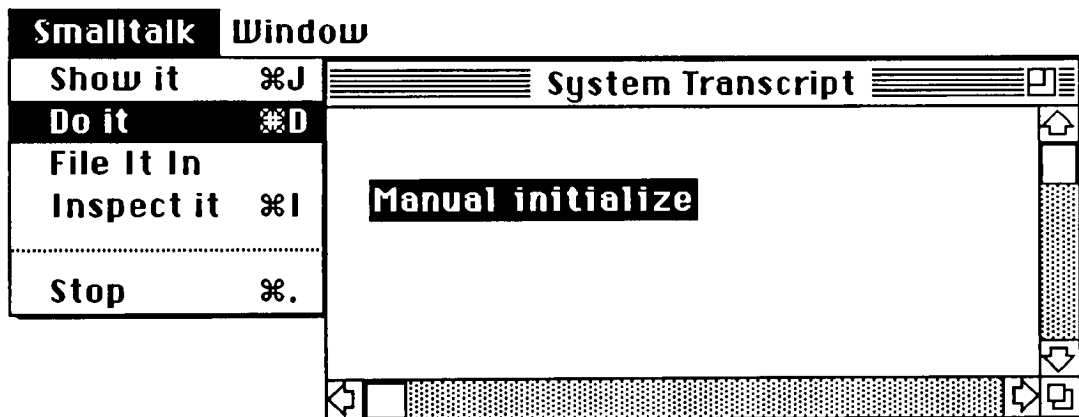
To Add an Annotation .....	53
Place the Annotation Mark .....	53
Enter the Annotation .....	53
Save the Annotation .....	53
To Cancel the Creation of an Annotation.....	54
Cancel Methods.....	54
To GoBack.....	54
To Open the Annotation Browser .....	54
To Change Preferences .....	54
Link Mode .....	55
General Information.....	55
To open the system in Link mode .....	56
Link Menu.....	56
To Link From a document Pane.....	57
Link Source Selection Overlap Dialog Box.....	57
To link from an entry in the Index .....	57
To link from an entry in the Table of Contents.....	57
To Link to a Text Document.....	58
To link to an Image Document .....	58
To Save a Link .....	58
To Show (Hide) Link Source 'Buttons' in a Document.....	59
To Follow a Link:.....	60
In Full Link Mode.....	60
In Simulated Browsing Mode.....	60
To Remove a Link .....	60
Browse / Link .....	60
To Change to Simulated Browsing Mode .....	61
To Change (Back) to Full Link Mode.....	61
Browse / Link Notes.....	61
Link Info Box .....	61
To Create an Info Box Link.....	61
File Menu.....	62
To Open a Text File.....	63
To Open an Image File .....	63
To Open the Table of Contents .....	63
To Open the Index .....	63
To Save Changes to the Index Entries List .....	63
To Save the Current State of the Hyperdocument Link Table .....	64
To File In the Link Table from a File .....	64
To File Out the Current State of the Link Table .....	64
To Close the Currently Active Window .....	64
Index Menu and Icon Ribbon.....	64
To Open Text into the Index.....	65
To Open an Image File .....	65
To Add an Entry to the Index List .....	65
To Add a Subentry to the Index List.....	66
To Remove an Entry or Subentry.....	66
To Save the Current State of the Index List.....	66

Appendix A. User Manual

**Initializing the System**

**Note:** The following process need only be done once for a given hyperdocument.

Open the Smalltalk/V image that contains the system and type `Manual initialize` in the transcript window. Select those two words and choose **DoIt** from the **Smalltalk** menu. You will be asked to choose a text file from the folder where the text files for the hyperdocument you want to open are located. Next you will be asked to select an image file from the folder where the image files for the hyperdocument are located. This operation sets up the file paths to the documents of the hyperdocument you want to open. [See Fig. 5]



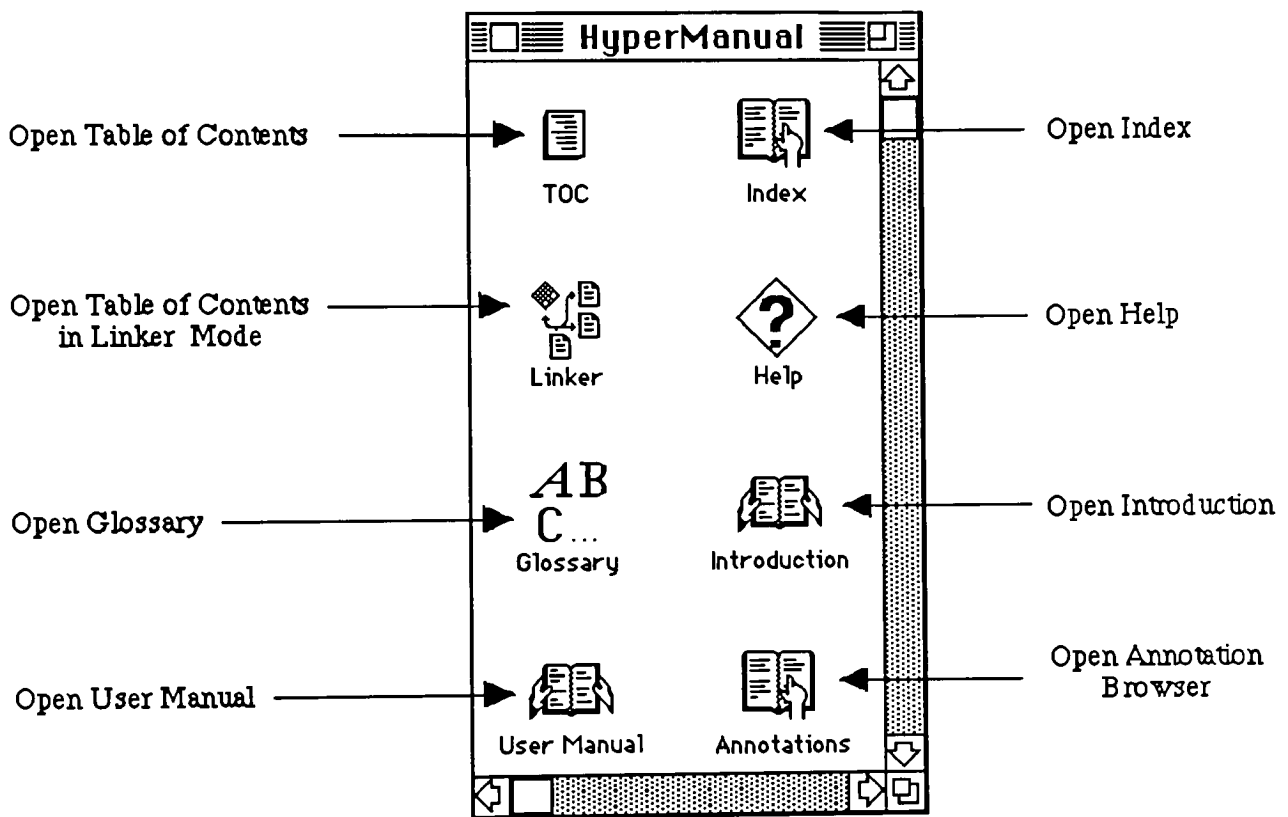
**Fig. 5 -- HyperManual Initialize**

**Start Up**

To open the hyperdocument, type `ManualIntroPane new open` in the transcript window. Select those three words and choose **DoIt** from the **Smalltalk** menu. This will open the *Tool Palette* for the system.

## Tool Palette

The tool palette contains subtitled icons that activate the various browsing and linking facilities in the system. The palette contains icons for the Table of Contents, Index, Linker, Glossary, Help, and Annotation Browser. There may also be icons for opening certain parts of the hyperdocument directly. To activate any of the facilities, click on the appropriate icon. All choices except Linker will open a view on the hyperdocument in browse mode. Choosing the Linker icon will open the Table of Contents in Link mode. [See Fig. 6]




**Fig. 6 -- Tool Palette**

## Browse Mode

In Browse mode, you can view a hyperdocument and annotate it, but you may not add or remove any of the basic links in the hyperdocument.

## Table of Contents

### To Open

To open the Table of Contents, a) if the tool palette is the active window, either click on the Table of Contents icon or choose the **Table of Contents** option from the **File** menu or b) if you are in one of the systems facilities (and in the Browse mode), choose **Table of Contents** from the **HyperText** menu or click on the  icon in the icon ribbon.

### Description and Usage

The Table of Contents window gives you quick access to all the sections of the hyperdocument. It is a two pane window containing the *entries pane* at the top, and a button pane at the bottom. The entries pane has the properties of a multilevel table of contents; that is, it shows access points to the hyperdocument at more than one level of detail. [See Fig. 7]

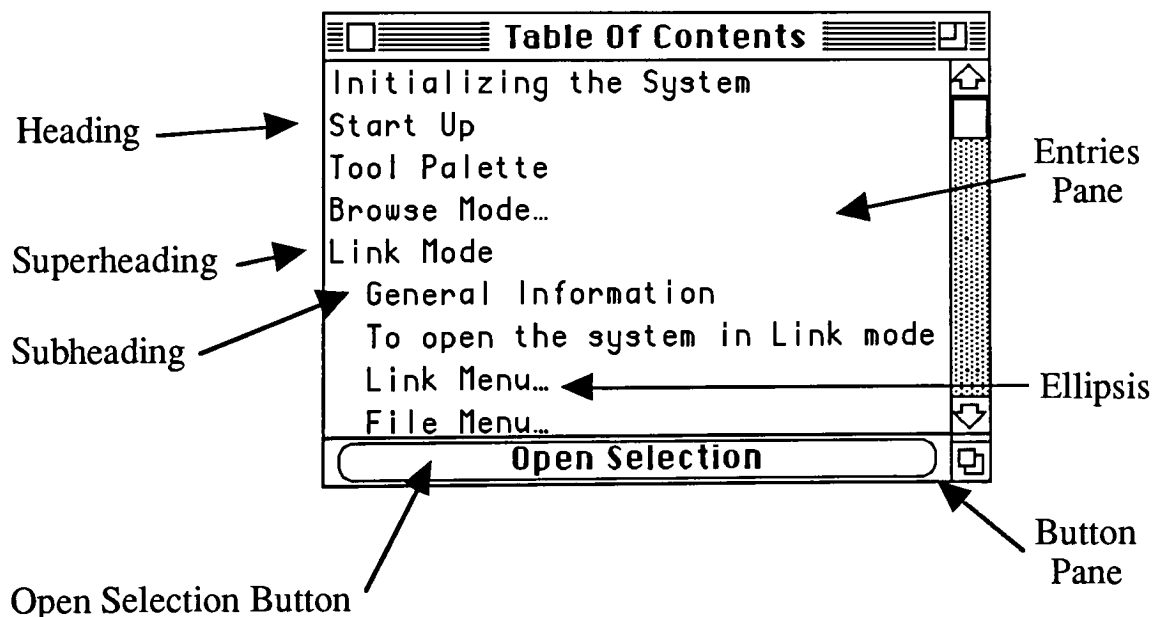


Fig. 7 -- Table of Contents

## Choosing a Heading


When the Table of Contents first opens, the entries pane contains a list of the first level headings in the hyperdocument. If a heading has a subheading, that fact will be indicated by an ellipsis following the heading. If you double-click on a heading with an ellipsis, the table of contents list will expand to include the immediate subheadings of that heading (and the ellipsis after the expanded heading will disappear). Each subheading will be indented two spaces relative to its superheading. Any subheading with an ellipsis may also be expanded. To hide subheadings, double-click on their superheading. To select a heading at any level, click on the desired heading with the mouse.

## Opening a Document Window

Select a heading in the entries pane in the Table of Contents. Click on the **Open Selection** button in the button pane at the bottom of the window. A document window containing the information in the selected heading will open.

## Index

### To Open

To open the Index, a) if the tool palette is the active window, either click on the Index icon or choose the **Index** option from the **File** menu or b) if you are in one of the systems facilities (and in the Browse mode), choose **Index** from the **HyperText** menu or click on the  icon in the icon ribbon.

### Description and Usage

The Index window gives you quick access to all the indexed information in a hyperdocument. The Index window has four panes: the entries pane, reference pane, icon ribbon pane, and document pane. [See Fig.8]

HyperManual  
Appendix A -- User Manual

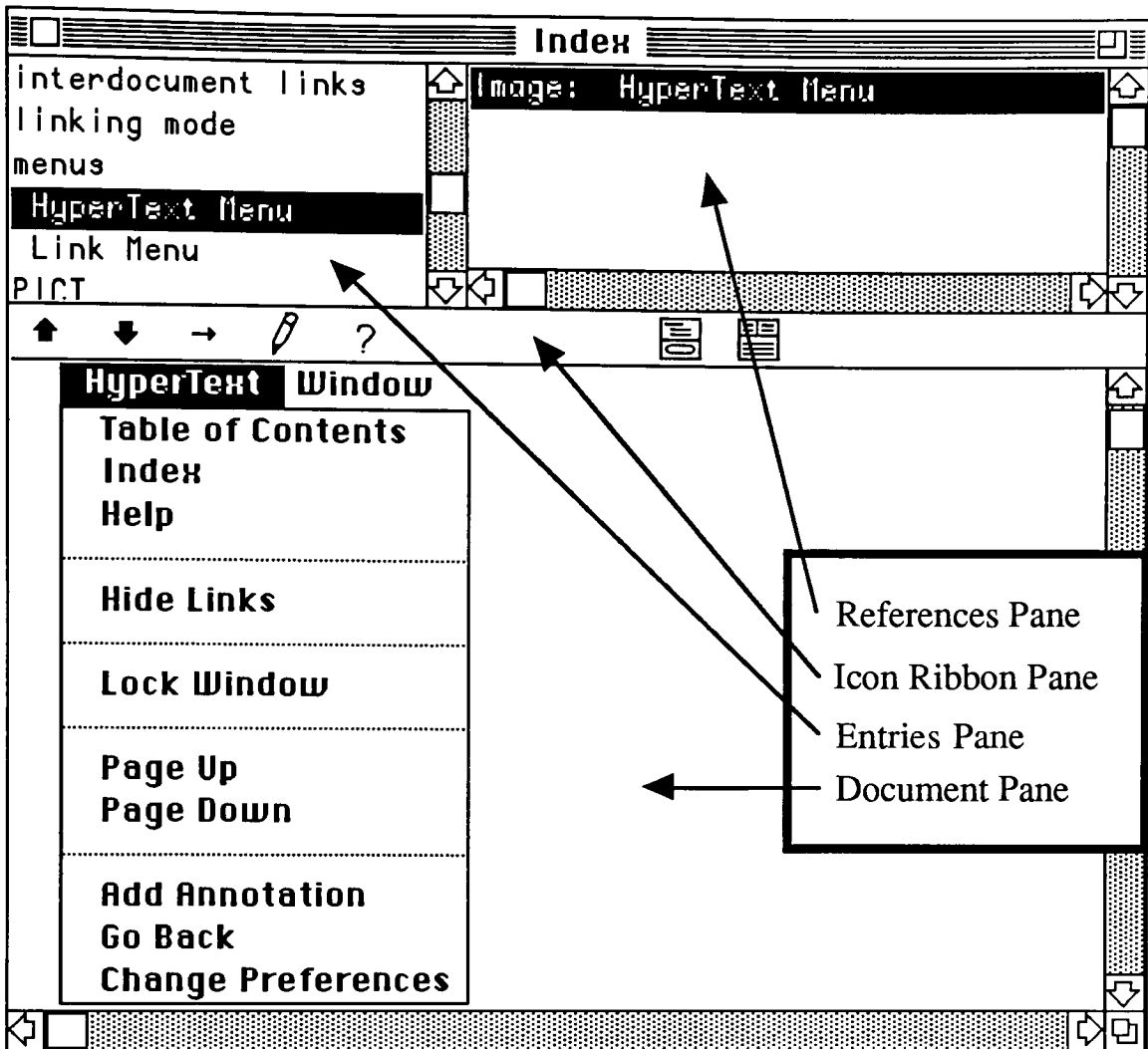


Fig. 8 -- Index Window

### Entries Pane

The entries pane is located in the upper left quadrant of the Index window. It contains an alphabetized list of the first level entries (topics) in the index. If an entry has a subentry, that fact will be indicated by an ellipsis following the entry. If you double-click on an entry with an ellipsis, the index list will expand to include the immediate subentries of that entry (and the ellipsis after the expanded entry will disappear). Each subentry will be indented two spaces relative to its superentry. Any subentry with an ellipsis may also be expanded. To hide subentries, you double-click on their superentry. To select an entry at any level, click on the desired entry with the mouse.

## References Pane

The references pane contains the references to the selected entry in the index. The references pane is located in the upper right quadrant of the Index window. If you click on (select) an entry in the entries pane, the references associated with the selected entry will appear as a list in the references pane. Each reference consists of two parts: 1) a one word descriptor of the kind of document the reference is located in and 2) a phrase that gives you some idea of the information to be found if you follow that reference. If you select a reference, the associated information will appear in the document window.

## Icon Ribbon Pane

The icon ribbon pane gives you quick access to most of the options in the **HyperText** menu. It appears as a horizontal strip that runs the width of the Index window about one third of the way from the top of the window. The icon ribbon contains eight icons: Page Up, Page Down, Show Links, Help, Add Annotation, Table of Contents, and Index. [See fig. 16] The functionality of the icons in the icon ribbon is described in the discussion of the **HyperText** menu.

## Document Pane

The document pane appears in the lower half of the Index window. If you select a reference in the reference pane, the associated document will appear in the document pane. If the document is textual, the referenced word or phrase will be highlighted. If the referenced document is an image, an appropriate portion of the image will be highlighted. The document pane in the Index window has the same functionality as a *Document Window*. For example, you can follow links out of the index document pane to an associated document window.



## Annotation Browser

### To Open

To open the Annotation Browser, a) if the tool palette is the active window, either click on the Annotation Browser icon or choose the **Annotation Browser** option from the **File** menu or b) if you are in one of the systems facilities (and in the Browse mode), choose **Annotation Browser** from the **HyperText** menu.

### Description and Usage

The Annotation Browser window gives you quick access to all the annotations you have added to a hyperdocument. The Annotation Browser window has six panes: the document list pane, button pane, references pane, annotation pane, icon ribbon pane, and document pane. [See Fig. 9]

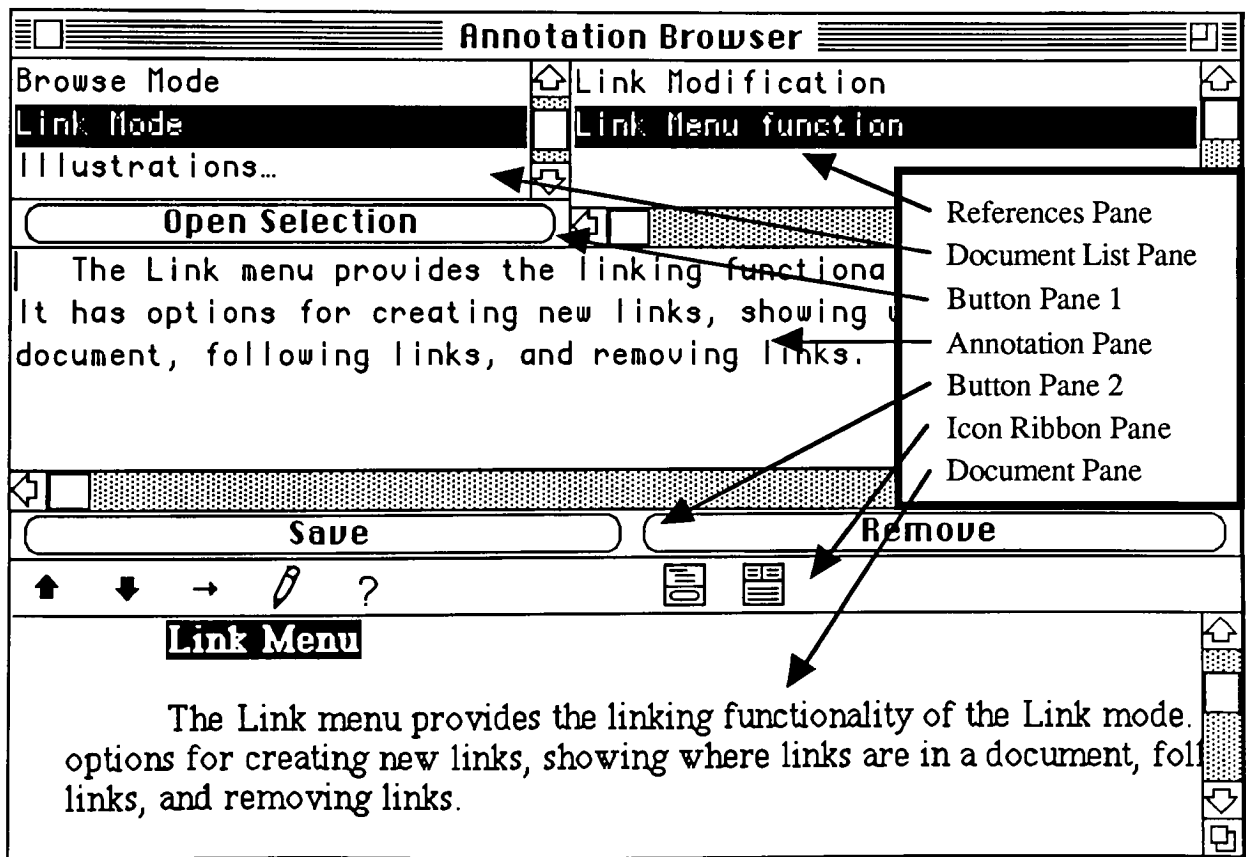


Fig. 9 -- Annotation Browser

## **Document List Pane**

The document list pane is located in the upper left quadrant of the Annotation Browser window. It contains a list of the documents in the current hyperdocument. This pane (with its associated button pane) behaves much the same way as the Table of Contents window behaves. The difference is that when you select an entry in document list pane and click on the open selection button, the selected document appears in the document pane of the Annotation Browser instead of in a new document window.

### **Button Pane 1**

The button pane is located immediately below the document list pane and contains a button titled **Open Selection**. If you click on the Open Selection button, the document selected in the document list pane will appear in the document pane.

## **References Pane**

The references pane contains the references to the entry selected in the document list pane. The references pane is located in the upper right quadrant of the Annotation Browser window. When an entry is selected in the document list pane the annotations associated with the selected entry will appear as a list in the references pane. Each reference consists of two parts: 1) the title of the document and 2) a phrase that was associated with that annotation. If you select a reference, the associated annotation will appear in the document annotation pane and the document in which that annotation was placed will appear in the document window.

## **Annotation Pane**

The annotation pane is located in the middle of the window. This pane contains the text of a new or existing annotation. To create or modify the text of an annotation, place the cursor where you want in the text pane and enter or change the text as desired.

## Button Pane 2

The button pane is located immediately below the annotation pane and contains two buttons: **Save** and **Remove**. The function of these buttons is explained in the description of the Annotation Window.

## Icon Ribbon Pane

The icon ribbon pane gives you quick access to most of the options in the **HyperText** menu. It appears as a horizontal strip that runs the width of the Annotation Browser window and is located just above the document pane. The functionality of the icons in the icon ribbon is described in the discussion of the **HyperText** menu.

## Document Pane

The document pane appears in the bottom third of the Annotation Browser window. To make a document appear in the document pane, first, select the document you want to view from the list in the document list pane. If no entries appear in the references pane, or if you simply want to view a document, open the document by clicking on the **Open Selection** button in the button pane. If there are entries in the references pane, you may select the annotation you want to see and the document will appear in the document window. If the document is textual, the text will be scrolled to the location of the annotation mark in the text and the text near the annotation will be highlighted. In an image, the annotation mark will be superimposed on the image. The document pane in the Annotation Browser window has the same functionality as a *Document Window*. For example, you can follow links out of the Annotation Browser document pane to an associated document window.

## Document Windows

There are two types of document windows; those that contain text and those that contain images. Most of the functionality of the two types is the same.

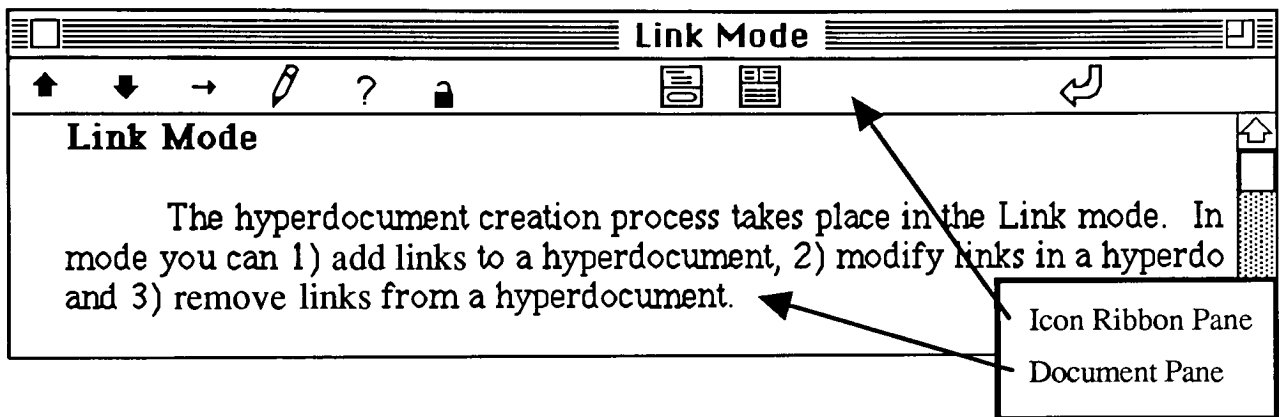
## To Open

There are three ways to open a document window:

- 1) Click on the appropriate icon in the tool palette.
- 2) Open a document from the Table of Contents window.
- 3) Follow a link from a document pane. [See below]

## Description and Usage

The document window is the basic view on the information contained in a hyperdocument. It has two panes: an icon ribbon pane and a document pane. [See Fig. 10]



**Fig. 10 -- Document Window**

### Icon Ribbon Pane

The icon ribbon pane gives you quick access to most of the options in the **HyperText** menu. It appears as a horizontal strip that runs the width of the document window just beneath the title bar. The functionality of the icons in the icon ribbon is described in the discussion of the **HyperText** menu.

### Document Pane

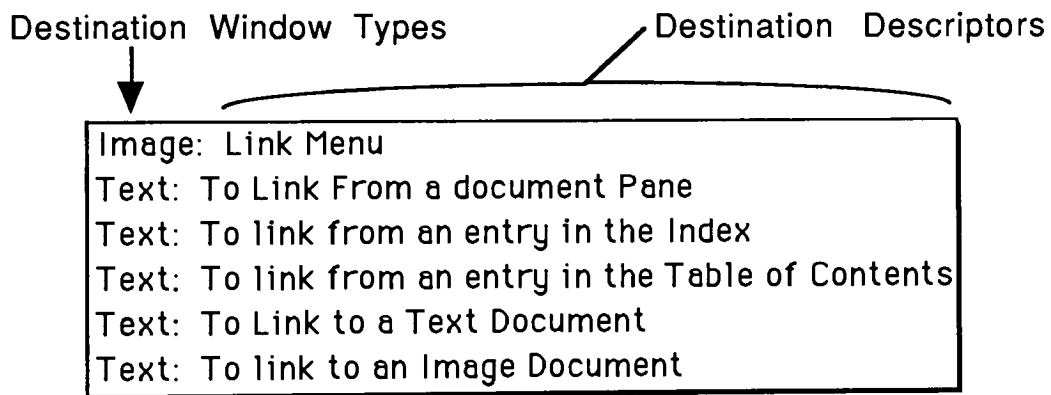
The document pane fills the rest of the document window below the icon ribbon. It contains either text or an image. The state of the document pane can be modified by using either the **HyperText** menu or the icon ribbon.

## To Follow a Link

Position the cursor within a link source region and click the mouse button. A *destination list window* will pop-up near the current position of the cursor. Click on the destination you want to go to and a document window will open containing the chosen destination.

### Destination List Window

A *destination list window* will pop-up near the current position of the cursor when you follow a link in a document. Each entry in the destination list consists of two parts: a one word descriptor for the type (Text, Image, or Note) of the destination document and a phrase that gives you some idea of the information to be found in that document. To follow a link, click on the destination you want to go to and a document window containing the chosen destination will open. [See Fig. 11]



**Fig. 11 -- Destination List Window**

## Note Window

The Note Window is very similar to the Note window in Guide. It contains brief textual explanation relevant to the selected item.

### Description and Usage

The note window has one pane and contains only text. [See Fig. 12]

A web is the set of links that define a hyperdocument.

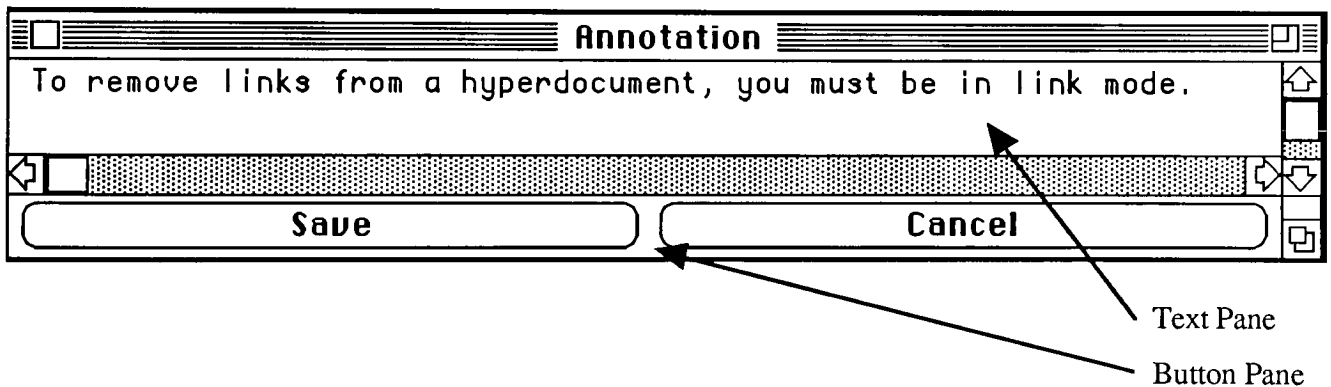
**Fig. 12 -- Note Window**

To open a note window, follow a link from a document pane to a *Note*. [See Browse Mode: Document Windows: ... To Follow a Link]

To close the window, click with the mouse anywhere outside the window.

### Annotation Window

There are two forms of Annotation windows, one for the creation of a new annotation and one for the viewing, modification, or removal of an existing annotation. (The second form adds the remove button to the button pane.) The annotation window has two panes, a text pane and a button pane. [See Fig. 13]



**Fig.13 -- New Annotation Window**

#### Text Pane

The text pane is located in the upper portion of the window. This pane contains the text of a new or existing annotation. To create or modify an annotation, place the cursor where you want in the text pane and enter or change the text as desired.

## Button Pane

The button pane contains either two or three buttons depending on the form of the annotation window. In the *annotation creation* form of the window, the button pane contains a **Save** button to save the annotations and a **Cancel** button which cancels the annotation process. The *annotation view, modify, remove* form of the window adds the **Remove** button which makes it possible for you to remove an existing annotation from a document in a hyperdocument. [See Fig. 14]

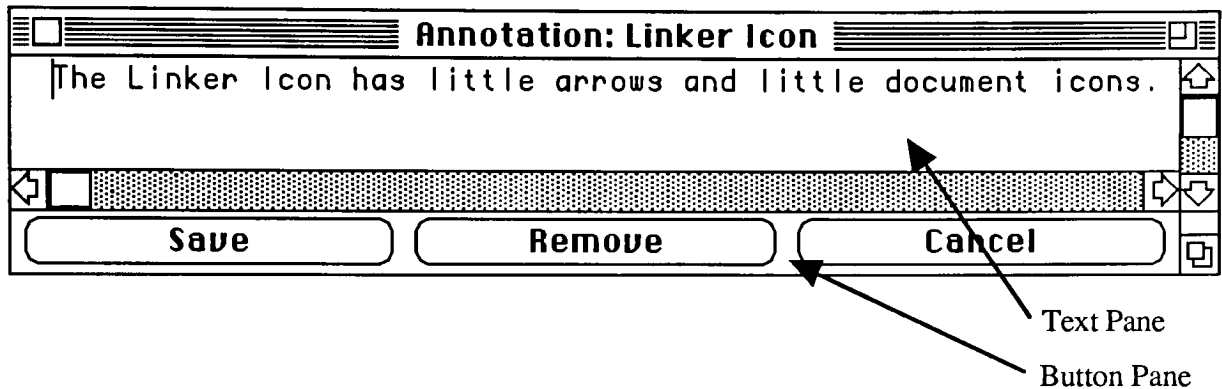


Fig. 14 -- Annotation Window

### To View an Annotation

In a document pane,

- 1) Click on the annotation mark of an annotation to open an annotation window on that annotation.
- 2) When you are through, click on the **Cancel** button

### To Modify an Annotation

In a document pane,

- 1) Click on the annotation mark of an annotation to open an annotation window on that annotation.
- 2) Make the changes you want in the text pane of the annotation window.
- 3) Click on the **Save** button to save the changes.

## **To Remove an Annotation**

In a document pane,

- 1) Click on the annotation mark of an annotation to open an annotation window on that annotation.
- 2) Click on the **Remove** button at the bottom of the annotation window.  
The annotation window will close and the annotation will be removed from the system.

## **HyperText Menu and Icon Ribbon**

You use the HyperText menu and icon ribbon to open certain system tools and to modify the state of the associated document pane. (In the HyperText menu for the Table of Contents, the document manipulation functions are not active.) The complete HyperText menu has eleven menu options and the complete icon ribbon has nine options. (The icon ribbon does not include an Open Annotation Browser icon or a Change Preferences icon.) Not all menu or icon options are available at all times. For example, the open Index option is inactive inside the index. The description that follows is based on the state of the HyperText menu and icon ribbon when they are associated with a document window that contains text. This state is chosen because it illustrates all the options available. (The following actions refer to figures fifteen and sixteen.)



HyperText Window
Table of Contents Index Help
Hide Links
Lock Window
Page Up Page Down
Add Annotation Go Back Change Preferences Annotation Browser

Fig. 15 -- HyperText Menu

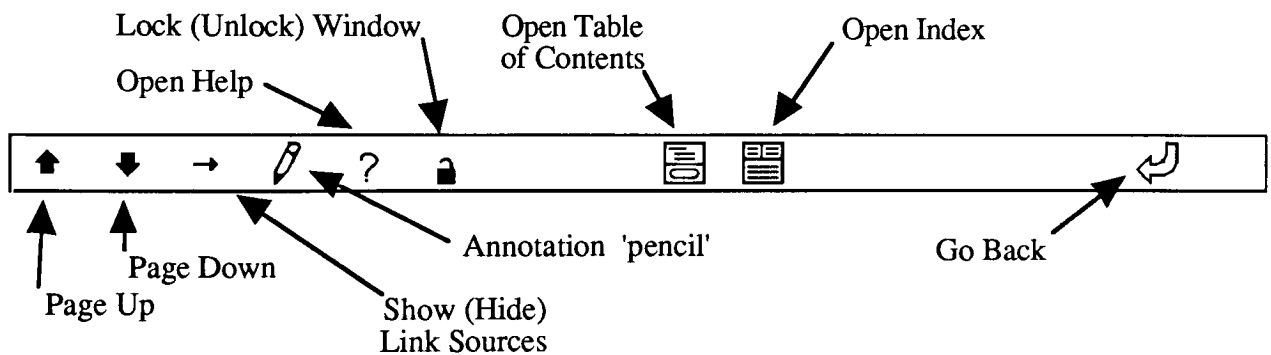




Fig. 16 -- Icon Ribbon

## Actions

### To Open or Activate the Table of Contents

- Choose Table of Contents from the HyperText menu or
- Click on the Table of Contents icon:  in the icon ribbon.

### **To Open or Activate the Index**

- a) Choose **Index** from the **HyperText** menu or
- b) Click on the **Index** icon:  in the icon ribbon.



### **To Open or Activate Help**



- a) Choose **Help** from the **HyperText** menu or
- b) Click on the **Help** icon: ? in the icon ribbon.

### **To Show (Hide) Link Source 'Buttons' in a Document**

- a) Choose Show (Hide) Links from the HyperText menu or
- b) Click on the Show (Hide) icon: → in the icon ribbon or
- c) Tap the space bar.


### **To Lock (Unlock) a Window**

**Purpose:** Locking (unlocking) a window prevents it from being (allows it to be) automatically closed by the system. The icon's state indicates whether the window is currently locked:  or unlocked: . To **Lock (Unlock)** a window,

- a) Choose **Lock (Unlock)** from the **HyperText** menu or
- b) Click on the **Lock (Unlock)** icon:  (  ) in the icon ribbon.


### **To Page Up**

**Purpose:** A text document may be opened on a portion of document. **Page Up** adds a page of text before the current first line of the text (if possible). To **Page Up**,

- a) Choose **Page Up** from the **HyperText** menu or
- b) Click on the **Page Up** icon:  in the icon ribbon.


### **To Page Down**


**Purpose:** A text document may be opened on a portion of document. **Page Down** adds a page of text after the current last line of the text (if possible). To **Page Down**,

- a) Choose **Page Down** from the **HyperText** menu or
- b) Click on the **Page Down** icon:  in the icon ribbon.

### **To Add an Annotation**

**Purpose:** The **Add Annotation** option makes it possible for you to add your own notes to any document in the hyperdocument. To **Add an Annotation**,

- a) Choose **Add Annotation** from the **HyperText** menu or
- b) Click on the **Add Annotation** icon:  in the icon ribbon.

The cursor will change to a pencil: . [Continue with Place the Annotation Mark below.]

### **Place the Annotation Mark**

Move the point of the pencil to the place in the document where you want to put the annotation mark for the annotation you are creating. Click the mouse button. An Annotation window will open.

### **Enter the Annotation**

Type the information you want into the text pane of the note window. When you are satisfied, click on the **Save** button in the button pane to save the note. The annotation window will close and a dialog box will appear. In the Dialog box, enter a word or phrase to remind you of the general contents of the note.

### **Save the Annotation**

To save the annotation, click on the **Accept** button in the dialog box (or hit the return key). Your note will be saved and linked to the document.

## To Cancel the Creation of an Annotation

The creation of a note may be canceled at any point in the creation process.


### *Cancel Methods*

- 1) To cancel note creation before you have positioned the annotation mark (while the cursor is shaped like a pencil), click with the mouse outside the document pane.
- 2) To cancel note creation after the annotation window is open, click on the **Cancel** button in the button pane of the annotation window.
- 3) To cancel note creation in the dialog box, click on the **Cancel** button.

In each case, you will return to the state you were in before you selected the annotation option.

## To GoBack

Purpose: The **Go Back** option activates the systems retrace facility. The *retrace* facility makes it possible for you to retrace the sequence of links you followed to get to your current location. Each time you choose the Go Back option, you go back one step in the link sequence. To **Go Back**,

- a) Choose **Go Back** from the **HyperText** menu or
- b) Click on the **Go Back** icon:  in the icon ribbon.

## To Open the Annotation Browser

Choose **Annotation Browser** from the **HyperText** menu. [See Browse Mode: Annotation Browser]

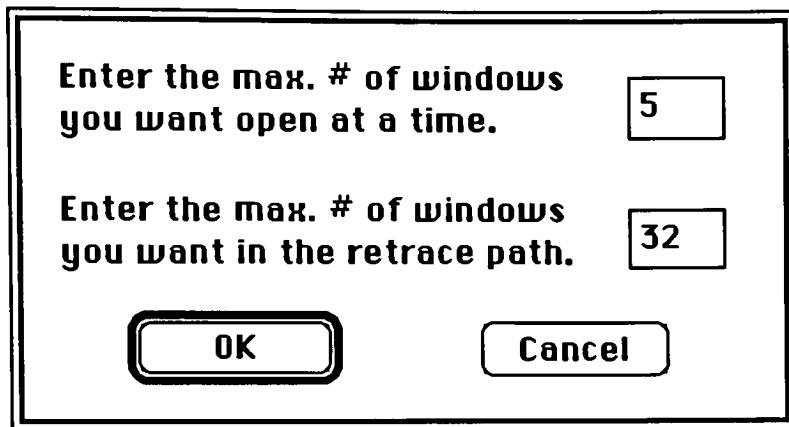
## To Change Preferences

Purpose: The **Change Preferences** option makes it possible for you to change 1) the maximum number of document windows you want open at a time and 2) the maximum number of steps stored in the retrace facility. To make a change,

## HyperManual

### Appendix A -- User Manual

- 1) Choose **Change Preferences** from the **HyperText** menu. A dialog box will appear. [See Fig. 17]



**Fig. 17 -- Change Preferences Dialog Box**

- 2) To change the value for either field, select the field you want to change (double-click on the desired box on the right side of the dialog box) and enter the new value.
- 3) a) Click on the **OK** button when (and if) you are satisfied with your changes or  
b) Click on the **Cancel** button if you decide you do not want to make any changes.

## Link Mode

The hyperdocument creation process takes place in the Link mode. In the Link mode you can add links to a hyperdocument, modify links in a hyperdocument, and remove links from a hyperdocument.

## General Information

All the functionality of the Browse mode of the system is maintained in the Link mode of the system. The only functional difference is in the process of following a link in a document pane. [See Link Mode: Link Menu: To Follow a Link]

In the **Link** mode, you can add links to, and remove links from, a hyperdocument. The basic process of creating a link involves 1) selecting the source of the link, 2) setting destination, 3) (for a textual destination) selecting the context of the link, and 4) saving the link. The linking operations (with the exception of saving the link) may be preformed in any order and may be intermixed with non-linking operations such as opening a file. (You may not however save a link until you have performed all the operations necessary to define the link.)

### To open the system in Link mode

- a) Click on the **Linker** icon in the Tools palette or [See Fig. 6]
- b) Choose the **Linker** option from the File Menu. [See Fig. 23]

A Table of Contents for the hyperdocument will open in Link mode.

### Link Menu

The **Link** menu provides the linking functionality of the Link mode. It has options for creating new links, showing where links are in a document, following links, and removing links. [See Fig. 18]

Link	Index	Window
Link From		⌘♦F
Link To		⌘♦T
Link Context		⌘♦C
Save Link		⌘♦S
Link Entry		⌘♦E
Link Info Box		
Show Links		⌘H
Follow Link		⌘Y
Remove Link		⌘♦R
Browse		

Fig. 18 -- Link Menu

## To Link From a document Pane

In a window that contains a document pane (a document window or the Index),

- 1) Select the region of an image or area of text that is to be the source of the link
- 2) Select **Link From** from the **Link** menu or enter command-shift-F at the keyboard. This establishes the selection as the source of the link.

## Link Source Selection Overlap Dialog Box

If the source selection for a new link overlaps an existing link source, a dialog box will open to indicate this problem. If you want to use the existing selection, click on the **Yes** button or hit the return key on the keyboard. If you are unsure of the link, or want to change the selection area, click on the **No** button. [See Fig. 19]

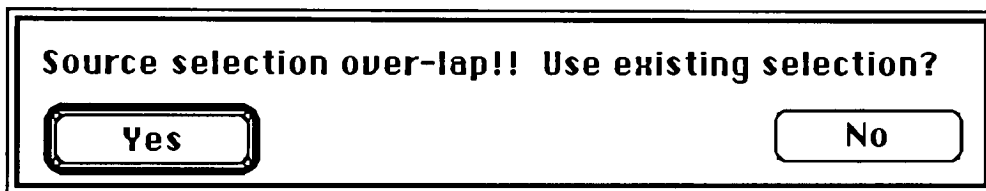


Fig. 19 -- Source Selection Overlap Dialog Box

## To link from an entry in the Index

In the Index Window

- 1) Select the (sub) entry that is to be the source of the link,
- 2) Choose **Link Entry** from the **Link** menu or enter command-shift-E at the keyboard. This establishes the selected entry as the source of the link.

## To link from an entry in the Table of Contents

In the Table of Contents Window

- 1) Select the (sub) entry [heading] that is to be the source of the link,

- 2) Choose **Link Entry** from the **Link** menu or enter command-shift-E at the keyboard. This establishes the selected entry [heading] as the source of the link.

### **To Link to a Text Document**

- 1) Create the link destination
  - a) Open the desired document if it is not already open. [See Link Mode: File Menu: To Open a Text File]
  - b) Select the word or phrase in the text that is to be the destination focus of the link.
  - c) Choose **Link To** from the **Link** menu or enter command-shift-T at the keyboard. This establishes the selected text as the link destination.
- 2) Create the Link Context
  - a) Open the desired document if it is not already open. [See Link Mode: File Menu: To Open a Text File]
  - b) Select the region of text (1 or more lines) that defines an appropriate context for the destination of the link.
  - c) Select **Link Context** from the **Link** menu or enter command-shift-C at the keyboard. This establishes the selected text as the link context.

### **To link to an Image Document**

- 1) Open the desired document if it is not already open. [See Link Mode: File Menu: To Open an Image File]
- 2) Select the region of the image that is to be the destination focus of the link.
- 3) Choose **Link To** from the **Link** menu or enter command-shift-T at the keyboard. This establishes the selected region as the link destination.

### **To Save a Link**

**Note:** You may only save a link that has its source and destination (and in the case of links to text, context) are properly defined.



HyperManual  
Appendix A -- User Manual

- 1) Choose **Save Link** from the **Link** menu or enter command-shift-S at the keyboard. A dialog box will appear. A default description of the link destination will be selected. You may wish to change or add to this description as it becomes the link explainer. [See Fig. 20]

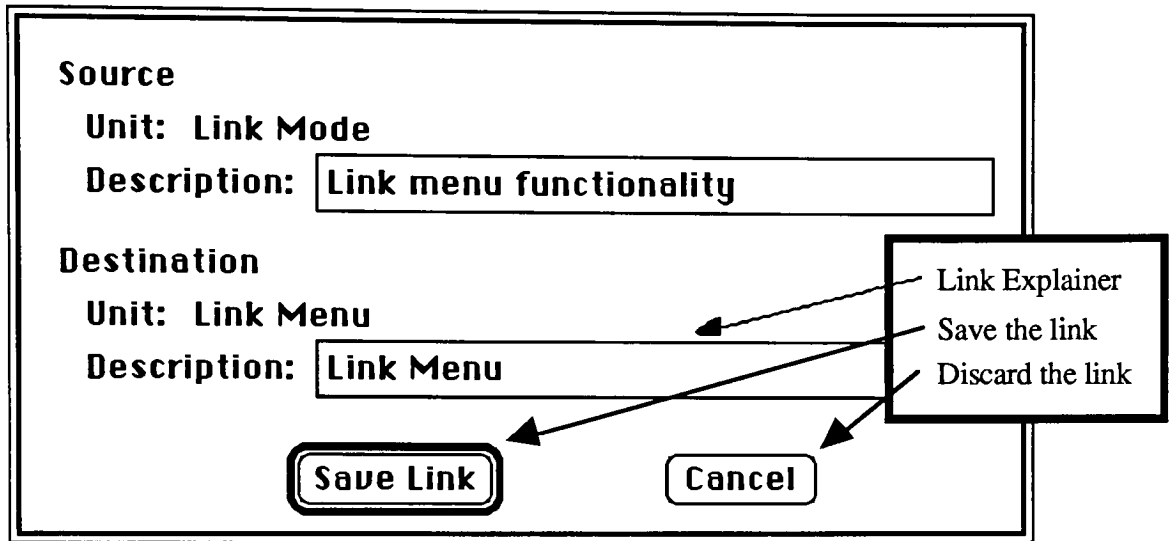


Fig. 20 -- Save Link Dialog Box

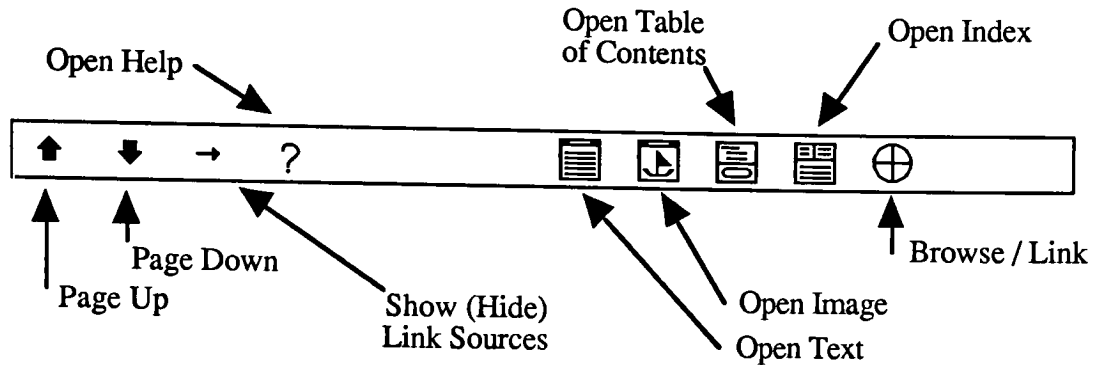
- 2) a) Click on the **Save Link** button when (and if) you are satisfied with the link. The new link will be saved.  
b) Click on the **Cancel** button if you decide you are not satisfied with the link.

**Note:** Once you have partially completed a link, you do not need to redo any part you are satisfied with, just redo the parts you are not satisfied with.

**To Show (Hide) Link Source 'Buttons' in a Document**

- a) Choose Show (Hide) Links from the Link menu or
- b) Click on the Show (Hide) icon: → in the icon ribbon or [See Fig. 21]
- c) Tap the space bar.

HyperManual  
Appendix A -- User Manual



**Fig. 21 -- Link Icon Ribbon**

**To Follow a Link:**

**In Full Link Mode**

- 1) Position the mouse cursor within a link source region. [See Link Mode: Link Menu: To Show (Hide) Links... to learn how to see link sources.]
- 2) Choose **Follow Link** from the **Link** menu or enter command-shift-Y at the keyboard.

**In Simulated Browsing Mode**

- 1) Change to simulated browsing mode. [See Browse / Link below]
- 2) Position the browse cursor: ⊕ within a link source region and click the mouse button.

**To Remove a Link**

- 1) Position the mouse cursor within a link source region. [See Link Mode: Link Menu: To Show (Hide) Links... to learn how to see link sources.]
- 2) Choose **Remove Link** from the **Link** menu or enter command-shift-R at the keyboard.

This will remove the selected link from the hyperdocument.

**Browse / Link**

The Browse option on the **Link** menu (the ⊕ icon on the icon ribbon) makes it possible for you to change to a simulated browsing mode while actually remaining in Link mode. The simulated browsing mode makes it possible to follow links in document panes just by clicking on a link source region.

## To Change to Simulated Browsing Mode

Choose **Browse** from the **Link** menu or click on the  $\oplus$  icon in the icon ribbon.

## To Change (Back) to Full Link Mode

Choose **Link** from the **Link** menu or click on the  $\curlywedge$  or  $+$  icon in the icon ribbon.

## Browse / Link Notes

- 1) The browse/link icon in the icon ribbon toggles between  $\oplus$  and  $\curlywedge$  (for text) or  $+$  (for images) as you toggle between link and browse mode.
- 2) When you are in link mode, the cursor will look like:  $\curlywedge$  if you are in a text document and  $+$  if you are in an image.
- 3) When you are in simulated browsing mode, the cursor will look like:  $\oplus$ .

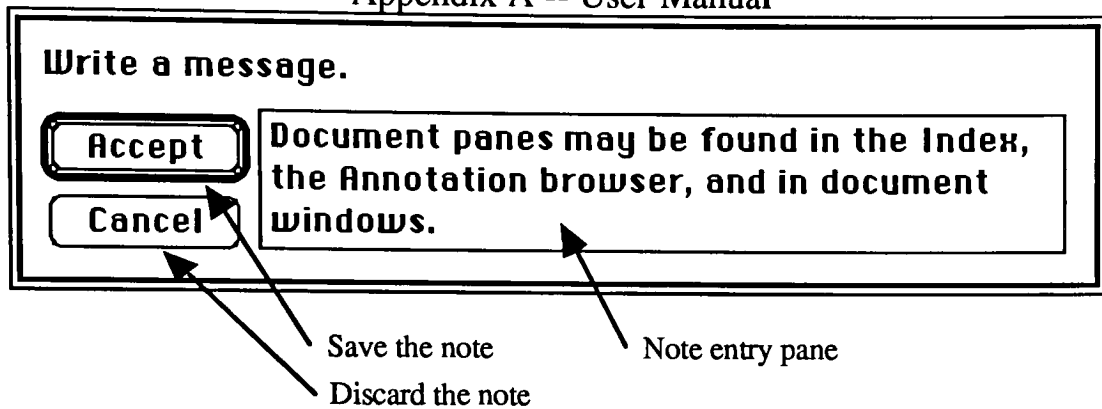
## Link Info Box

The **Link Info Box** option makes it possible for you to create a link to a pop-up information box. A pop-up information box is used to add short definitions, comments, etc., to a hyperdocument. [See Browse Mode: Note Window]

## To Create an Info Box Link

In a window that contains a document pane (a document window or the Index),

- 1) Select the region of an image or area of text that is to be the source of the link,
- 2) Choose **Link Info Box** from the **Link** menu. A dialog box will appear. [See Fig. 22]
- 3) Enter the information you want the user to see when the info box opens.
- 4) a) Click on the **Accept** button to save the link or  
b) Click on the **Cancel** button to cancel the creation of the link.



**Fig. 22 -- Info Dialog Box**


## File Menu


The **File** menu gives you access to the file manipulation functions needed to create a hyperdocument. In addition, it makes it possible for you to open (or activate if they are already open) the Index and Table of Contents. See figure 23 as a reference for the following menu option descriptions.

File	Edit	Smalltalk
Open Text		⌘O
Open Image		⌘⇧O
Open TOC		
Open Index		
Save Index		
Save Links		
File In Links		
File Out Links		
Close		⌘W
Quit		⌘Q


**Fig. 23 -- File Menu**


## **To Open a Text File**

Choose **Open Text** from the **File** menu or click on the  icon in the icon ribbon or enter command-o at the keyboard. A standard Macintosh file dialog box will appear. Find the directory that contains the RTF files for the hyperdocument you are creating and open the file you want. The text will appear in a new document window whose title will be the name of the file you opened.


**Note:** If the file you are opening has not previously been filed into the hyperdocument, it may take some time for the text to file in. If this is the case, a ticking watch cursor:  will appear to indicate that the system is working.

## **To Open an Image File**


Choose **Open Image** from the **File** menu or click on the  icon in the icon ribbon or enter command-shift-O at the keyboard. A standard Macintosh file dialog box will appear. Find the directory that contains the PICT files for the hyperdocument you are creating and open the file you want. The image will appear in a new document window whose title will be the name of the file you opened.

**Note:** It may take some time for the text to file in. If this is the case, a ticking watch cursor:  will appear to indicate that the system is working.

## **To Open the Table of Contents**

Choose **Open TOC** from the **File** menu or click on the  icon in the icon ribbon.

## **To Open the Index**

Choose **Open Index** from the **File** menu or click on the  icon in the icon ribbon.

## **To Save Changes to the Index Entries List**

Choose **Save Index** from the **File** menu.

### To Save the Current State of the Hyperdocument Link Table

Choose **Save Links** from the **File** menu.

### To File In the Link Table from a File

Choose **File In Links** from the **File** menu.

### To File Out the Current State of the Link Table

Choose **File Out Links** from the **File** menu.

### To Close the Currently Active Window

Choose **Close** from the **File** menu or enter command-w at the keyboard.

## Index Menu and Icon Ribbon

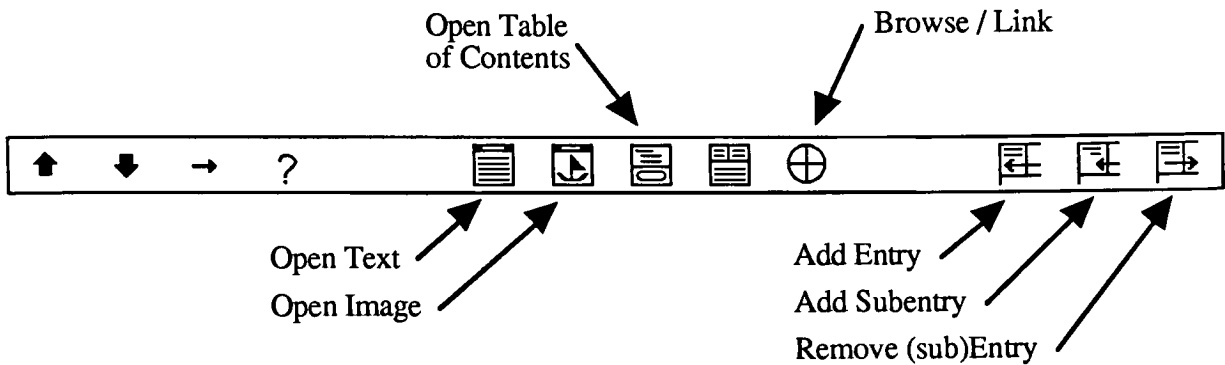
The **Index** menu gives you the special options needed to open document files in the index's document pane and to add and delete entries in the entries pane.  
[See Fig. 24 and 25]

Index	Window
Open Text	⌘T
Open Image	⌘I
<hr/>	
Add Entry	⌘E
Add SubEntry	⌘S
Remove Entry	⌘R
<hr/>	
Save Index	

Fig. 24 -- Index Menu


# HyperManual


## Appendix A -- User Manual




**Fig. 25 -- Index Icon Ribbon**


### To Open Text into the Index

Choose **Open Text** from the **Index** menu or click on the  icon in the icon ribbon or enter command-option-t at the keyboard. A standard Macintosh file dialog box will appear. Find the directory that contains the RTF files for the hyperdocument you are creating and open the file you want. The text will appear in the document pane of the index.


**Note:** If the file you are opening has not previously been filed into the hyperdocument, it may take some time for the text to file in. If this is the case, a ticking watch cursor:  will appear to indicate that the system is working.

### To Open an Image File

Choose **Open Image** from the **Index** menu or click on the  icon in the icon ribbon or enter command-option-i at the keyboard. A standard Macintosh file dialog box will appear. Find the directory that contains the PICT files for the hyperdocument you are creating and open the file you want. The image will appear in the document pane of the index.

**Note:** It may take some time for the text to file in. If this is the case, a ticking watch cursor:  will appear to indicate that the system is working.


### To Add an Entry to the Index List

- 1) Choose **Add Entry** from the **Index** menu or click on the  icon in the icon ribbon or enter command-option-e at the keyboard.

- 2) In the dialog box that pops up, type the entry you want to add to the index.
- 3) a) Click on the **Accept** button in the dialog box to complete the operation **or**  
b) Click on the **Cancel** button in the dialog box to cancel the operation.


The entry will be inserted into the index entries list at the alpha-numerically correct position.

### **To Add a Subentry to the Index List**

- 1) Select the entry (or subentry) that is to be this entries superentry.
- 2) Choose **Add SubEntry** from the **Index** menu or click on the  icon in the icon ribbon or enter command-option-s at the keyboard.
- 2) In the dialog box that pops up, type the subentry you want to add to the index.
- 3) a) Click on the **Accept** button in the dialog box to complete the operation **or**  
b) Click on the **Cancel** button in the dialog box to cancel the operation.

The subentry will be inserted into the index entries list at the alpha-numerically correct position under its superentry.

### **To Remove an Entry or Subentry**

- 1) Select the entry (or subentry) that you want to remove.
- 2) Choose **Remove** from the **Index** menu or click on the  icon in the icon ribbon or enter command-option-r at the keyboard.

### **To Save the Current State of the Index List**

Choose **Save Index** from the **Index** menu.



Classes created for the HyperManual system are **bolded**.

## Dispatchers

```
Object
  Dispatcher
    ScrollDispatcher
    GraphDispatcher
    ImageDispatcher
    ImageLinkerDispatcher
    ListSelector
    ModifiableListDispatcher
    TextEditor
    FormattedTextEditor
    HyperTextEditor
    HyperLinkEditor
  TopDispatcher
    HyperTopDispatcher
```

## Panes

```
Object
  Pane
    SubPane
    GraphPane
    ImagePane
    ImageLinkPane
    TextPane
    FormattedTextPane
    HyperTextPane
    HyperLinkPane
  TopPane
    HyperTopPane
```

## **Models**

### **Pane Support Models**

#### **String**

Object

StringModel

FormattedStringModel

**HyperStringModel**

#### **Selections**

Object

**ImageSelection**

**MultipleSelection**

**MultiImageSelection**

**MultiTextSelection**

TextSelection

FormattedTextSelection

**HyperTextSelection**

## **HyperMedia Models**

- Object**
  - Collection**
    - IndexedCollection**
    - OrderedCollection**
    - Path**
  - HyperAddress**
    - ImageAddress**
    - IndexAddress**
    - NoteAddress**
    - SimpleTextAddress**
    - TableOfContentsAddress**
    - TextAddress**
  - Link**
  - Manual**
  - WindowStruct**

## **Window Model Classes**

- Object**
  - Annotation**
  - HyperImage**
  - HyperText**
  - ManualIntroPane**
  - TableOfContents**
    - Index**
    - AnnotationBrowser**

HyperManual  
Appendix C -- Glossary

**Appendix C. Glossary**

active window	the top most window, the window with its title bar filled in with horizontal lines
annotate	the term for the creation of links in the Intermedia system
block	source and destination locations of a link in the Intermedia system
bookmark	a user controlled 'mark' on a window that keeps the window from being automatically closed by the system
browsing mode	the mode in the HyperManual system that makes it possible for the user to follow links in and add annotations to a hyperdocument but not create new general links
click	press and release the mouse button without moving the mouse
dialog box	a box that pops-up on the display when an application needs more information from the user
document window	a window on a portion of the hyperdocument that may contain text or images
double-click	click twice rapidly with the mouse button
entry	a topic in the Index of the HyperManual system
entriesPane	a ListPane in the Table of Contents, Index, or Annotation Browser in the HyperManual system
explainer	information tied to a link that explains to the users something about the information available if the link is followed

HyperManual  
Appendix C -- Glossary

Guide	a hypertext authoring and browsing system for the PC from Owl International, Inc.
Guideline	a hypertext document created in Guide
heading	the title for a page, chapter, etc., in a hyperdocument
hyperdocument	a document that consists of a web and its associated documents
icon	a picture that is used to represent an object or action
icon ribbon	see icon ribbon pane
icon ribbon pane	a pane (usually associated with a document pane) that contains icons that makes it possible for the user to bypass choosing certain menu options
inactive	(menu or icon ribbon option) when an option is inactive, the user may not be able to choose it or it may have no effect
interdocument links	direct document to document links
linking mode	the mode in the HyperManual system that makes it possible for the user to create new links in and remove links from a hyperdocument
PICT	Macintosh name for a PICTure resource file
pop-up	a window that 'pops-up' usually near the position of the cursor that gives information relevant to the current position of the cursor
subentry	a subtopic in the Index of the HyperManual system

## HyperManual

### Appendix C -- Glossary

superentry	a main topic or heading that has subheadings in the Index of the HyperManual system
subheading	the title or heading of a subdivision of a document
superheading	the main title or heading of a document that has subheadings
title bar	the strip at the very top of a window that contains the title of the window
tool palette	contains subtitled icons that activate the various browsing and linking facilities in the HyperManual system
web	the set of links that define a hyperdocument