Context and Orientation in Hypermedia Networks

KENNETH UTTING and NICOLE YANKELOVICH
Brown University

The core of hypermedia's power lies in the complex networks of links that can be created within and between documents. However, these networks frequently overwhelm the user and become a source of confusion. Within Intermedia, we have developed the Web View—a tool for viewing and navigating such networks with a minimum of user confusion and disorientation. The key factors in the Web View's success are a display that combines a record of the user's path through the network with a map of the currently available links; a scope line that summarizes the number of documents and links in the network; and a set of commands that permit the user to open documents directly from the Web View.

Categories and Subject Descriptors: H.3.4 [**Information Storage and Retrieval**]: Systems and Software

General Terms: Human Factors

Additional Key Words and Phrases: Hypermedia systems, hypertext systems, network browsers

1. INTRODUCTION

Hypermedia has the potential to greatly improve reading, writing, teaching, and learning, but it also has the potential to dramatically confuse and confound readers, writers, teachers, and learners. The promise of hypermedia is the ability to produce complex, richly interconnected, and cross-referenced bodies of multimedia information. Unfortunately, hypermedia also has the ability to produce complex, disorganized tangles of haphazardly connected documents.

At the Institute for Research in Information and Scholarship, one of our ongoing goals for the IRIS Intermedia [28] environment is to add tools to help authors and browsers manage the inherent complexity of a network of connected ideas. Conklin [7, p. 40] breaks the problem down into two parts:

—*disorientation*: the tendency to lose one's sense of location and direction in a nonlinear document; and
—*cognitive overhead*: the additional effort and concentration necessary to maintain several tasks or trails at one time.

Funding for this research was sponsored in part by Apple Computer, Inc. and by IBM.
Authors' address: Institute for Research in Information and Scholarship, Brown University, P.O. Box 1946, Providence, RI 02912; kdu@iris.brown.edu, ny@iris.brown.edu.

In this paper, we describe how a portion of the Intermedia system was redesigned to provide better orientation cues and reduce cognitive overhead.

IRIS Intermedia includes a word processor, a structured graphics editor, a bitmap viewer, and a timeline editor. In Intermedia, users are able to create these various types of documents (text, graphics, etc.) in a manner quite similar to most other Macintosh® applications. In addition, though, a user may create a *block*, a selectable portion of a document. Blocks are connected by *links* to form a network of inter- and intra-document connections. These connections are stored in a database, called a *web*. The same documents may contain different blocks and links in any number of webs. All links in Intermedia are bidirectional. That is, if you can follow a link from Block A to Block B, then you can also always follow it from Block B to Block A.

Along with these applications, Intermedia furnishes users with a set of tools for organizing documents and viewing the web. An integrated Finder facility provides users with a conventional hierarchical file organization. As with the standard Macintosh Finder, nodes of the hierarchy are called *folders* and may contain documents and other folders. In addition, the user may access a particular network of links by opening a web. The occurrence of a link within a document is indicated to the user by a *link marker* icon. Finally, the *Web View* facility gives the user some sense of context or location in the web. These terms are illustrated in Figure 1. Intermedia's Web View is discussed later, but is omitted from the scenario that follows in Section 2.

In field trials [1, 21], Intermedia has been used as a browsing, annotating, and communication environment for college students, as a writing environment for authors of a medical textbook, and as a collaborative design environment for software developers. For example, professors in English, Biology, Anthropology, English as a Second Language, and Geology have prepared webs for their students connecting essays, illustrations, scanned images, and timelines. In these courses, students browse the materials to prepare for class or study for exams. They write analyses, linking their arguments to the primary data provided by the professor, and they complete assignments that often involve creating a set of connections between existing materials and documents of their own creation. In some of the courses, students are encouraged to link comments on each other's written work and communicate with the professor about their own work. The authors of the medical textbook and software developers formed small teams of professionals to use Intermedia for creating and updating documents meant to be read on-line as well as in printed format. Webs created in Intermedia have typically contained several hundred documents and have been as large as 1000 documents and 2000 links.

2. SCENARIO

We present a hypothetical scenario to illustrate the problems of disorientation and cognitive overhead.

Susan is a new insurance salesperson at Bailee's Insurance Company and is using a collection of materials stored in Intermedia to learn about her new company and job.

® Macintosh is a trademark of Apple Computer, Inc.

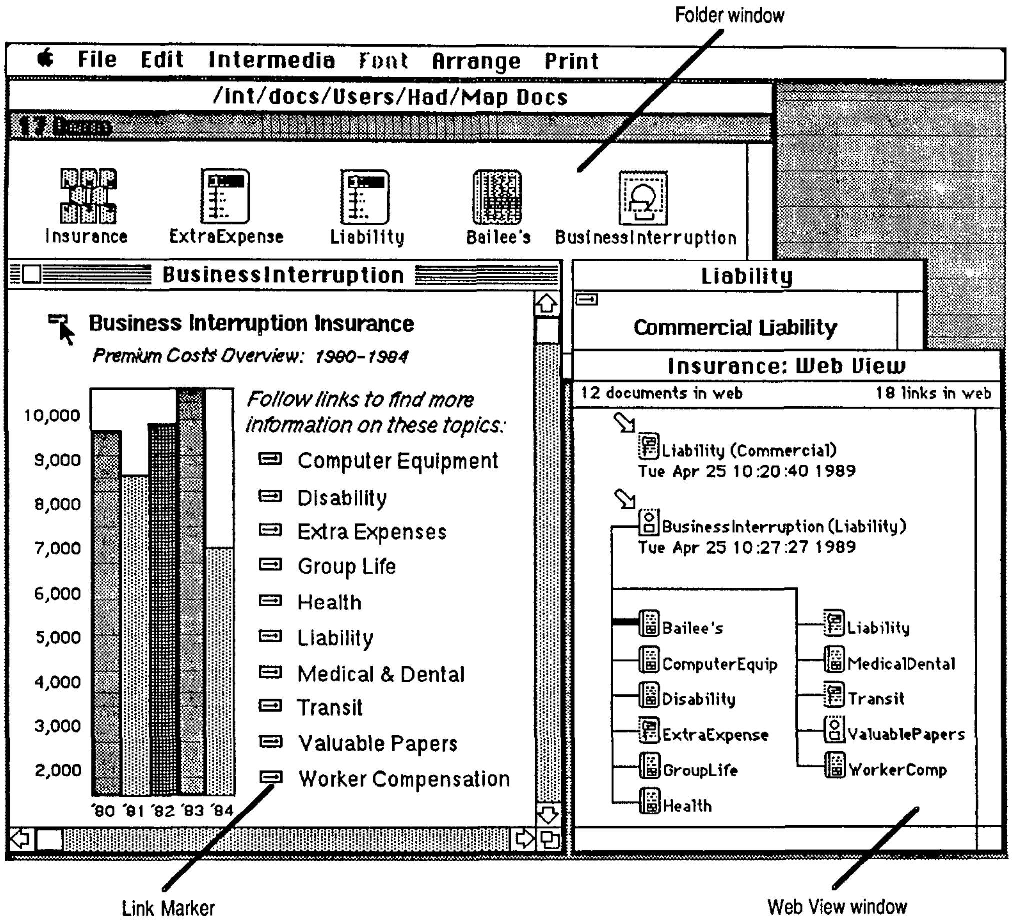

Fig. 1. Intermedia screen.

Susan opens the *Insurance* web and then a text document named *Bailee's*. She skims through a description of the company's history and finds a section that describes the various types of insurance Bailee's offers. A paragraph on disability insurance contains a link marker. Susan selects the link marker and chooses the Follow command from a menu at the top of the screen. Another text document, this one named *Disability*, appears on the screen in a window, partially overlaying that of the previous document, *Bailee's*. Susan skims through this document and decides to follow another link, which opens a third document, *GroupLife*, containing information about group life insurance.

By this point, Susan faces several problems. First, some of the links in *GroupLife* may lead her back to *Bailee's* or *Disability*. For the moment, however, Susan is only skimming the material and would prefer not to return to those documents. It is easy to envision similar problems in other situations, where users want to either avoid or limit themselves to certain documents, either on the basis of the document's name or type. For example, they may want to see only text or only graphics documents. To further complicate the situation, as Susan opens more documents, it will get harder to remember which documents have already been opened.

Another aspect of this problem is that Susan may want to go to all the documents connected to *GroupLife*, but following every link may be unnecessarily tedious. Several links may go to different portions of the same document, so once Susan follows one of these links and reads the entire document, following the other links to it is superfluous.

A second problem Susan faces is time. She begins to wonder how long it will take to get through all the material in the web. She wants to read *GroupLife* in more detail than the rest of the documents because she is attending a meeting on a client's group life policy at one o'clock, but she thinks it is even more important to have at least browsed the entire web before lunch. Without knowing how much material is in the web, Susan is unsure whether she should stop browsing and read *GroupLife* completely or continue browsing and come back at the end if she has enough time.

Trying to get back to *GroupLife*, once she has gone to another document, is Susan's third problem. There are a number of methods Susan can use to return. For example, *GroupLife* may still be open in some window on the screen. In this case, she can reactivate it simply by clicking on the window. However, the window may be completely overlaid by other windows, in which case those would have to be closed or moved in order to click on *GroupLife*. In fact, if the window is overlaid, Susan may not even realize that it is there. Additionally, leaving a lot of documents open quickly clutters the screen.

Another strategy Susan can take is to reopen *GroupLife* using the Finder. This is a fairly common approach and is often quite satisfactory. However, Susan opened *GroupLife* the first time by following a series of links, not by using the Finder. She may have no idea where *GroupLife* is located in the file system. In a large web of several hundred documents, this becomes a serious issue.

A third approach would be for Susan to retrace the steps she takes after she leaves *GroupLife*. For example, if she follows a link from *GroupLife* to *Liability* and from there to *BusinessInterruption*, she can follow those two links in the opposite direction to return to *GroupLife*. Alternatively, Susan can try to repeat the steps that brought her to *GroupLife* in the first place. That is, opening *Bailee's*, following the link to *Disability*, and from there to *GroupLife*. However, remembering or writing down the precise steps taken during a session is a rather tedious, if not impossible, task.

These three problems—determining where a link leads, estimating the size of the web, and returning to a document—all demonstrate aspects of both disorientation and cognitive overhead. The first and third problems demonstrate disorientation in Susan's confusion about where to go or, having decided on a destination, how to get there. The overhead occurs in the work she must do to accomplish these tasks; for example, remembering which links she follows so that she can retrace her steps. The second problem demonstrates disorientation in the sense that Susan is unaware of the boundaries of the information space she is exploring. Users cannot know that they are in the middle of the woods unless they know where the woods end. Again, Susan encounters cognitive overhead in the work she must do to help her guess at the boundaries.

3. GOALS

As mentioned earlier, Intermedia provides users with a Web View, which is intended to help ease the problems of disorientation and cognitive overhead. Early versions of the Web View were rather limited in the assistance they offered, so we embarked on a project to create an improved version.

Intermedia was initially designed with three kinds of Web Views: a global map, a local map, and a local tracking map. The global map portrayed every document in the web and the links between them. Figure 2 shows a global map for a very small web. Documents in the web are represented by their names drawn beneath icons indicating their document type: InterDraw (graphics), InterVal (timeline), InterWord (text) or InterPix (bitmap). The map shows only documents and cannot show the blocks within the documents. Because of this, the map draws a single line between any pair of documents that contain blocks that are linked to each other, no matter how many links the documents share.

Although the global map worked well for a small web, in a typical web of several hundred documents and a similar number of links, such a map was much too large and entangled to be of any use. The local maps were more useful. The local map showed a particular "focus" document specified by the user, and the documents to which it was linked, as shown in Figure 3. The local tracking map was exactly the same as a local map, except that it updated its focus dynamically as the user opened and activated documents. The original design called for the user to be able to display one or more of any of these types of views for an open web.

The decision was made, however, to implement only a single view of the web rather than giving the user a choice of the three styles. The style chosen initially, the global map, primarily served as an effective demonstration of "how not to do it," and was quickly abandoned. The local tracking map was more understandable but lacked any real functionality and so was usually ignored by users. Although these tools were sufficient for users to create and navigate rather complex webs, as our field trials have demonstrated, it was equally clear that there was plenty of room for improvement.

In the early versions of Intermedia, users relied on manual methods for a general organization of documents; they stored strongly related documents in the same folder and named folders and documents in an illuminating manner. The structure of important subsets of the web was frequently portrayed graphically in an *overview document* drawn by the creators of the webs using the InterDraw application (see Figure 4). These organizing strategies reflect a substantive understanding of the meaning of the contents of the web, and as such are valuable tools, but their necessary reliance on manual effort by authors and users limits their potential role.

We felt that the Intermedia system itself should provide more effective tools for handling webs. Disoriented users need context information to reestablish a sense of location. In particular, two types of context were needed: spatial context that answers the question "where can I go from here?" and temporal context that answers the question "how did I get to here?" In our first implementation, the local tracking maps provided the spatial information, but not very effectively, and the system did not provide the temporal information at all.

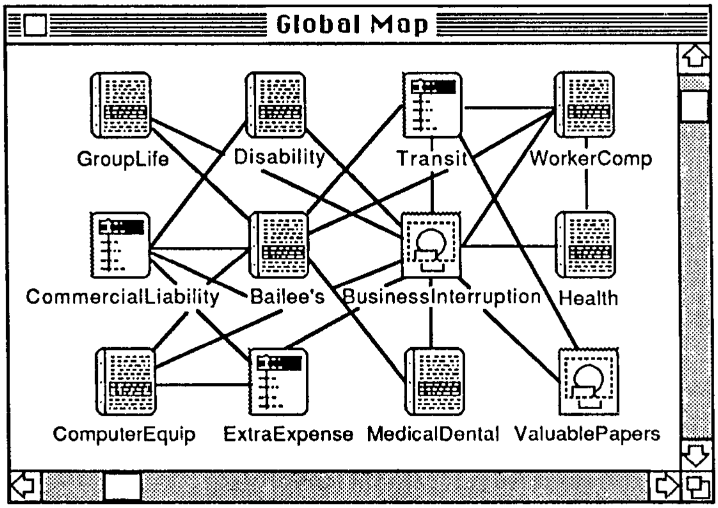

Fig. 2. Global map.

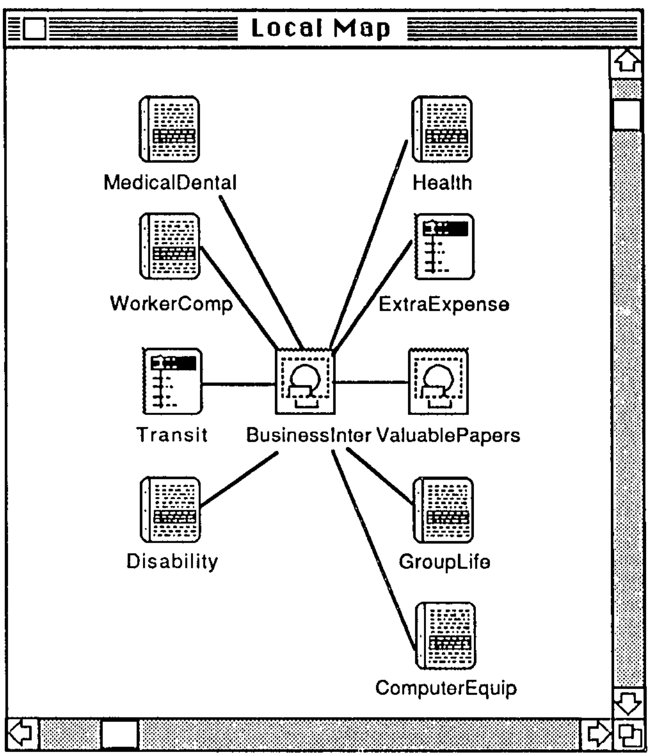

Fig. 3. Local map.

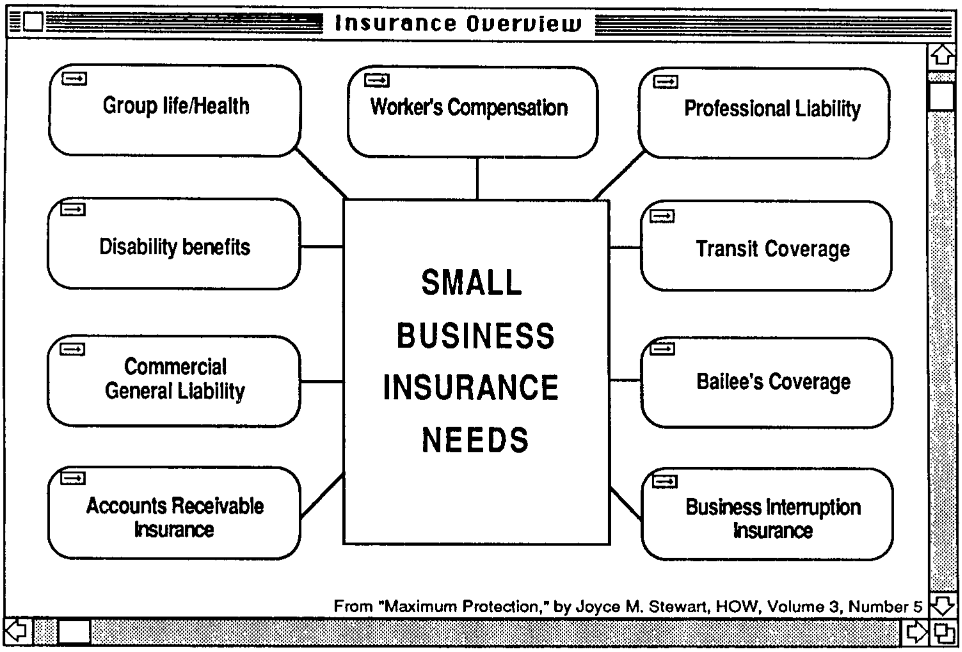

Fig. 4. Overview document.

Presenting the contextual information in a nonintrusive manner was another priority. Some of the systems described in the next section, for example, require or encourage the user to explicitly assign spatial location to nodes of the hypertext network. Since Intermedia users already assign documents to folders, we did not feel it was appropriate to further burden them in this manner. Some systems also require the user to explicitly request information about the network, such as a local map for the current document. This too we sought to avoid. The user's primary goal, after all, is to create and/or assimilate the information in the web, not to expend effort manipulating the web itself.

A final goal of a new design was to provide the information to the user in a compact manner that made efficient use of screen space. As in most windowing systems, screen space in Intermedia is a highly valued commodity, and we felt that the amount devoted to viewing the web should be kept to a minimum, allowing the rest of the screen to be allocated to the various documents the user has open. Not only should the display be compact, it should be flexible enough to adapt itself to whatever screen space the user allots to it.

4. OTHER APPROACHES

During the process of redesigning our Web View, we examined the manner in which other systems provide users with spatial and temporal context.

4.1 Approaches Relating to Spatial Context

Palenque [26] allows children to explore information about Mayan ruins using a schematic "you-are-here" map of the physical archaeological site and another map of a floor plan of a museum. All the information contained in the Palenque database is associated with physical locations—either site coordinates or rooms in the museum. For this read-only application, the visualization of data in terms of physical location is extraordinarily effective. The problem with this physical layout approach for more general-purpose hypermedia systems is twofold. First, these maps are composed once: all the data to be included has been established and then the maps are drawn to reflect some finite data set. Second, meaningful maps based on physical locations are almost always dependent on the content of the material, the individual user's angle of interest, and on the innate topographic qualities of the subject. Although there is a class of applications that might make effective use of geographical-type maps and/or floor plans, there are far more applications for which this type of map would be completely inappropriate.

The Palenque site map attempts to represent a global view of the data. Of all the systems we reviewed, it is probably the most effective global view, but this is due in part to the subject matter and in part to the read-only nature of the database. How could the system dynamically update the site map as data are changed? Another effective approach to representing a global view of connections in a database is exemplified in the *Learning Support Environment* [18]. In this environment, data are stored hierarchically, thereby allowing the data to be collapsed into a global view that fits, in its entirety, on the screen. Although a hierarchical organization of data allows for easy compaction, is much more generic than a physical organization, and can accommodate addition of new data, it is typically not adequate for representing a nonhierarchical network of connections.

The *Electronic Document System (EDS)* [11, 12, 13] provides two types of displays of the network. In EDS, authors create "pages" of information that are clustered into a hierarchy of "chapters." Each chapter can contain any number of pages and subchapters. EDS consists of two separate components: an authoring system, called the Document Layout System, and a browsing system, called the Document Presentation System. In the Document Layout System, authors can open multiple windows, each containing a different chapter anywhere in the hierarchy. Links between pages stored in the same chapter are always displayed within these windows. Links between pages in different chapters are displayed only on request. Lines are drawn across window boundaries to show connections.

This scheme provides a *semiglobal* view of the link network, meaning that subsets of the data are compacted and represented as a set rather than as individuals. The Document Presentation System is a read-only environment in which users could browse documents. Unlike Intermedia, links in EDS are unidirectional only.

The major advantage to EDS's semiglobal view is that a link can be drawn to or from a chapter rather than the page itself. Abstracting data in this way makes it more feasible to show connections at the global level. The disadvantage, however, is that the abstraction is not always meaningful. That is, knowing that there exists a link from somewhere in one subhierarchy to somewhere in another subhierarchy may be very useful sometimes, but completely useless at other times. Additionally, because the same chapter or its higher level chapters may appear in several windows, several link lines may be drawn that are actually duplicates, misleading the user about how many links actually emanate from a page or chapter. Furthermore, the display often fails to directly answer user questions about whether or not a particular pair of documents are linked, or exactly which pages are linked to a particular page of interest. Finally, drawing links across multiple windows tends to be space intensive.

EDS's Document Presentation System uses a display of the network, called "Neighbors," that differs from that of the Document Layout System. The Neighbors display is similar to Intermedia's local maps, except that, due to the directionality of EDS's links, all the preceding pages were displayed in a single column to the left of the "focus" page, and all the following pages were in a single column to the right of the focus page. This display suffers from the same problems as Intermedia's local maps, as well as the problem that the Neighbors display could not be viewed at the same time as the document itself.

Thoth-II [6] uses a different mapping approach. Rather than generating a global map at the outset of a session, the map is created dynamically as a user browses through linked nodes. Its "Spider" diagram technique, as shown in Figure 5, has a central node and all nodes are attached to it. Each link line is labeled. When the user clicks on a different node, the diagram then expands that node to show the nodes connected to it. The expansion creates new instantiations of all the connected nodes, rather than reusing ones already in the diagram. Consequently, link lines do not cross one another. However, as the user browses, the diagram quickly gets larger and larger. Tools are provided for scrolling, but this approach is tremendously space intensive.

Other hypermedia systems also provide global or semiglobal data views. *NoteCards* [16, 17] and *gIBIS* [8] provide graphical browsers, shown in Figures 6 and 7, of the entire network. Users are able to scroll these views as well as rearrange the placement of nodes. Particularly in the gIBIS system, users are encouraged to move new nodes from their default position to make the browser representation more meaningful. Both systems provide a feature for viewing the contents of the browser at multiple levels of detail. If the network is large, the highest level of detail shows the structure of the information, but no semantic information. The user can zoom in to see any portion of the browser in detail, but owing to space limitations, can never see the entire network in detail or in any compacted format that retains semantic information [14]. One useful aspect of these large browsers is that they give the user, at a glance, an idea of the size of their network. Users can tell roughly how many documents they are working with and how interconnected they are.

To solve the size problem, NoteCards and also *Neptune* [9] allow users to generate filtered browsers based on a query. In NoteCards, for example, a user can filter out information based on link type or node type. Browsers in Neptune are always constructed on the basis of content queries. If the query is broad enough, then the browser might show a global view of the entire network. If the query is well refined, however, the browser is likely to be a manageable size.

Neptune has an additional feature that helps users understand how a particular document of interest fits into the overall linked database. When Neptune users generate a browser based on a query, they are able to specify a target document. The browser is then drawn with that target document as the "root" [9]. This approach suffers from a standard database query problem: If users do not know

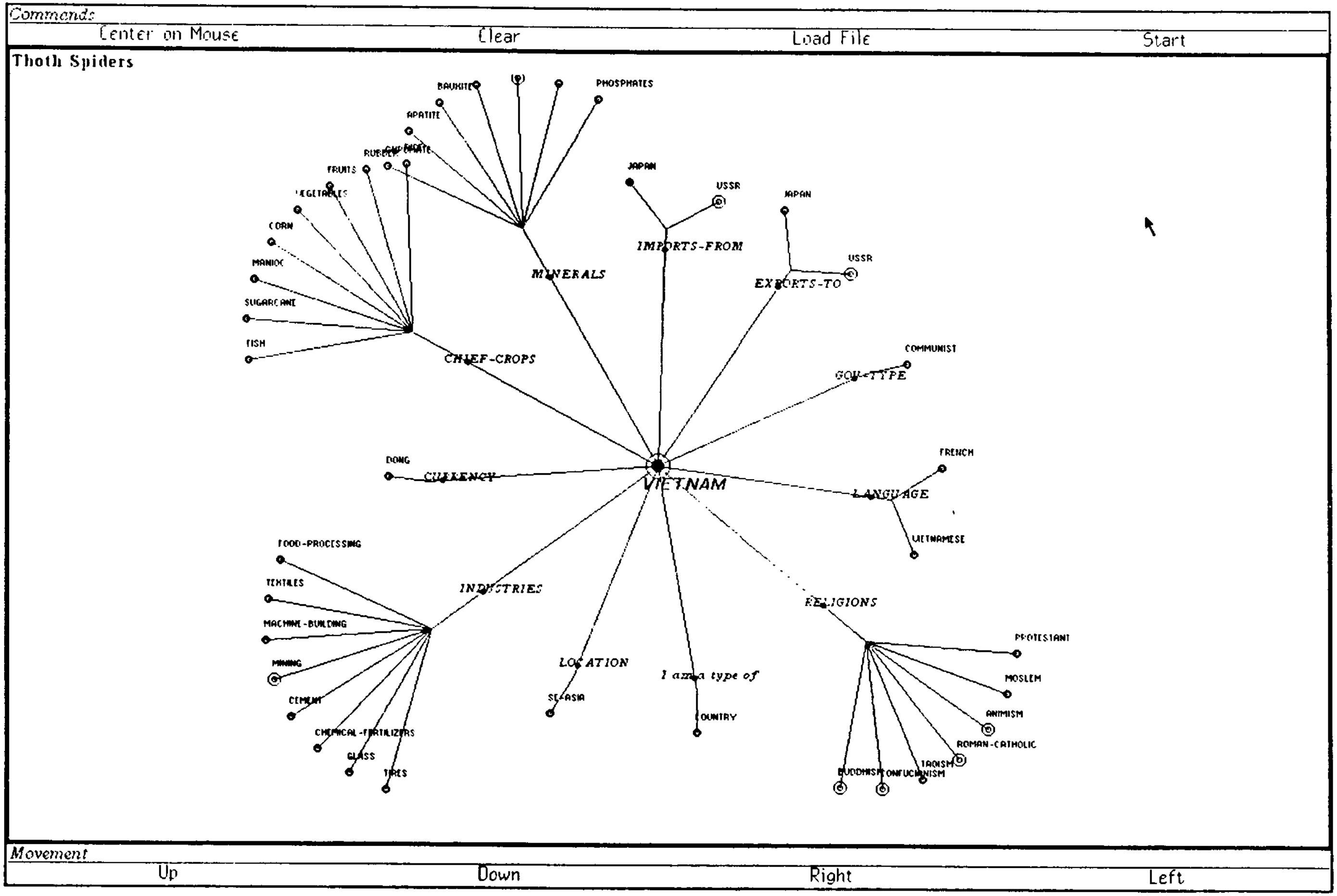

Fig. 5. Example of a Thoth-II Spiders diagram.

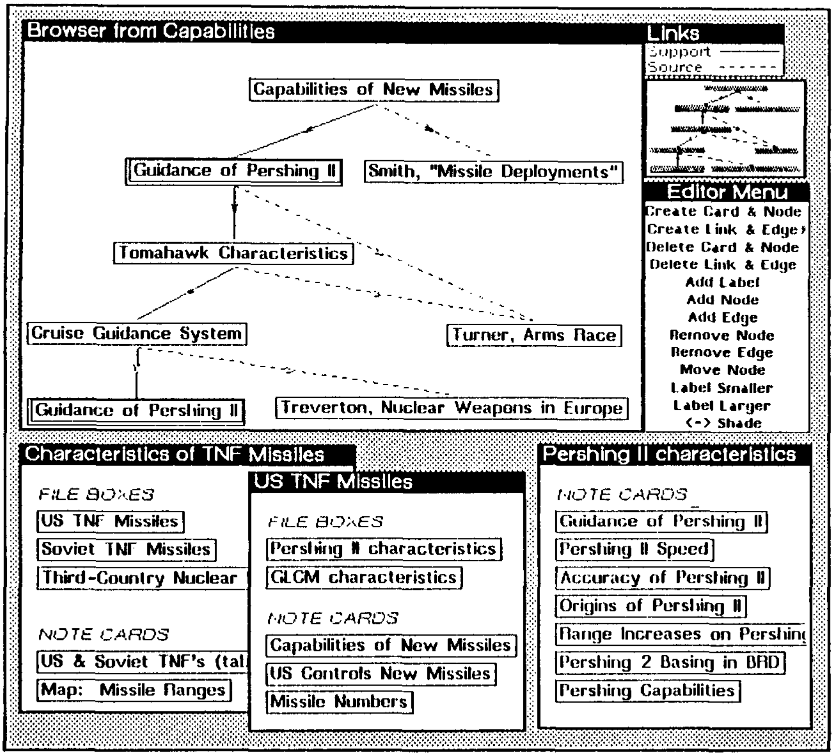

Fig. 6. A sample screen from NoteCards.

the right questions to ask, they may miss important information. In addition, Neptune, NoteCards, and gIBIS require that users continually reissue a query every time a new browser is desired.

4.2 Approaches Relating to Temporal Context

Two basic approaches have been used in the past to provide users with temporal context: backtracking and maintenance of a history trail.

Backtracking was a staple feature in early hypertext systems such as *FRESS* [5, 27]. This feature can also be seen in a number of currently available commercial systems. For example, *Guide* [19] and *HyperCard* [20] both provide users with a "back" button or key. By clicking on a button on the screen or pressing a key on the keyboard, users are able to step back, one node at a time, through their current session. In some cases, this feature proves to be invaluable, but it can also cause a degree of disorientation. These systems have no operation, such as "next," which undoes the effect of "back." Therefore, while you can travel backward through your session, you may have trouble retracing your steps back to the node from which you began your backtracking digression.

To alleviate this problem, HyperCard supplements the backtracking feature with a graphical history display called "Recent." This display contains miniatures, arranged in a grid, of the last 42 nodes visited. If a node is visited more than one time, it is not added to the display, so unlike the backtracking feature, the Recent

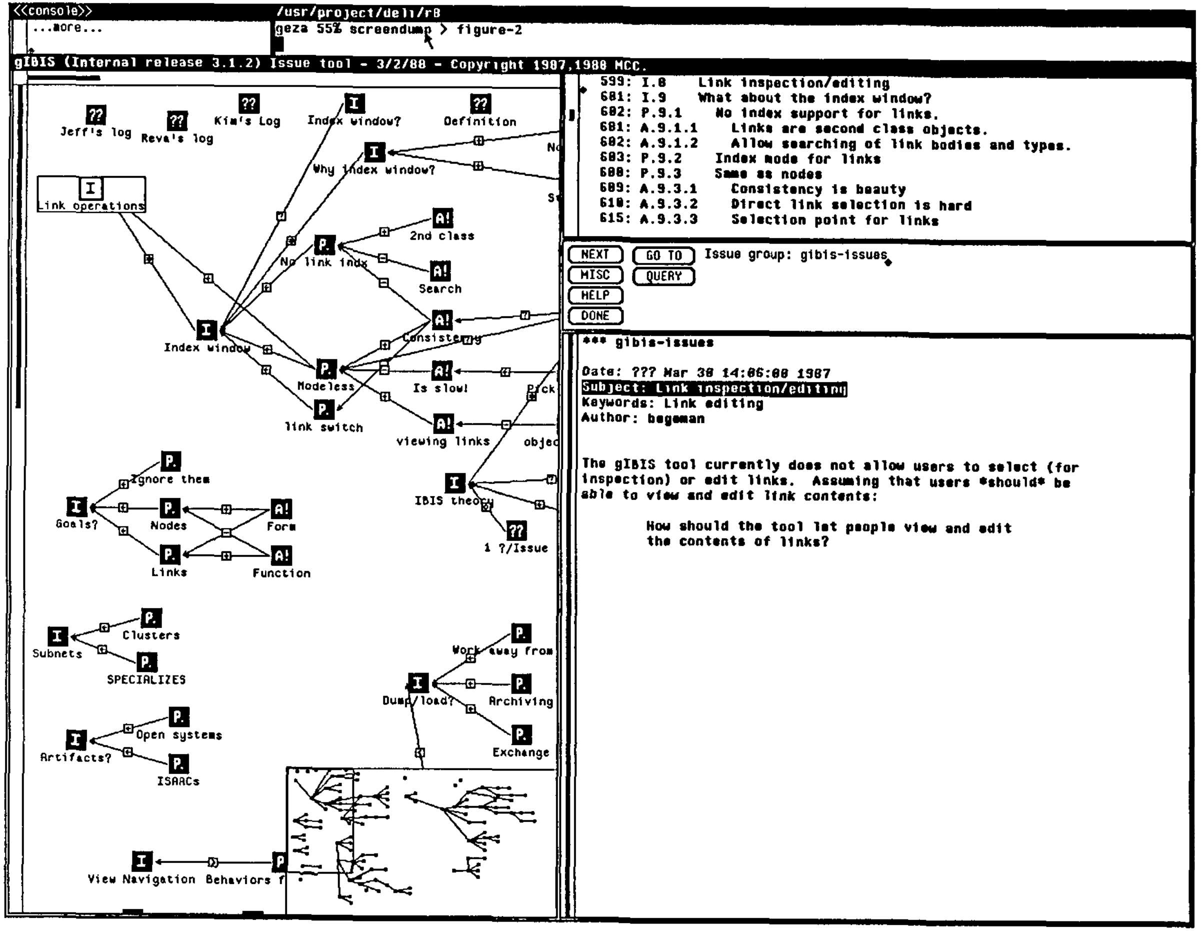

Fig. 7. The gIBIS interface.

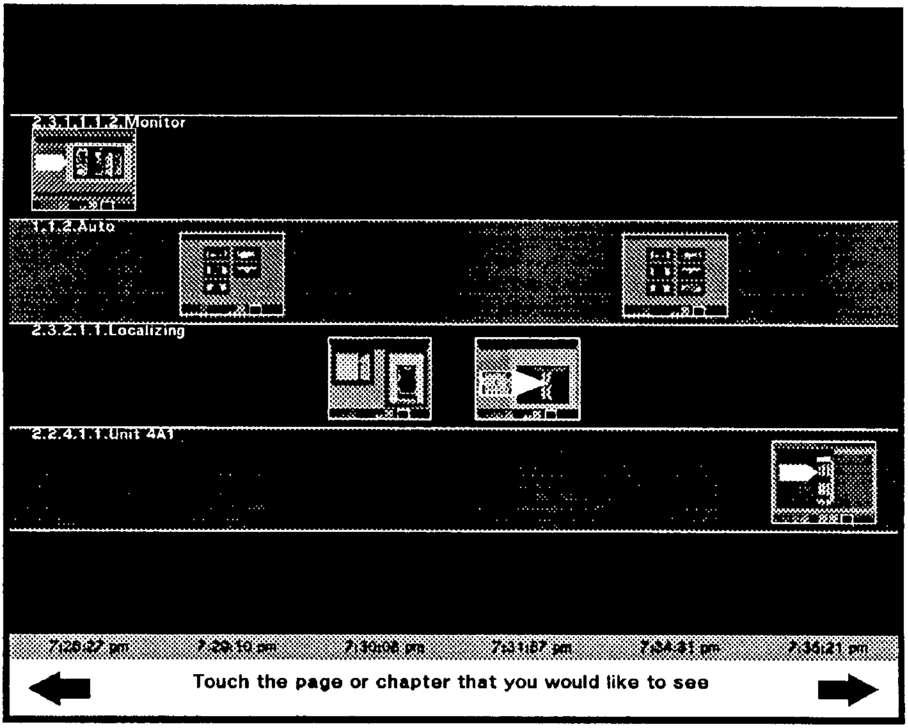

Fig. 8. An EDS timeline.

display does not contain a strictly historical trace. Clicking on any miniature in the display returns you to that node. The idea behind this sort of display is to take advantage of the user's visual memory. The notion is that you might not remember the name of the node you wish to return to, but you may remember what it looked like. EDS contains a somewhat more sophisticated graphical history display, illustrated in Figure 8. While it also made use of miniatures, each node is named and timestamped and maintained in exact order. A node visited more than once is repeated on the "Timeline." Like HyperCard, EDS allows users to select a miniature and return to the node represented by it [27].

Document Examiner [25] provides two types of historical trails, a command history and a history of "records" examined (usually a chapter of user documentation). The command history allows users to recapture any command previously issued during a session. In a separate window, Document Examiner creates a "Bookmark" for each record the user has opened. A list of Bookmarks is maintained and can be saved and reopened in a future session.

4.3 Providing Spatial and Temporal Cues in Concert

Foss [14] developed a set of extensions to NoteCards aimed specifically at solving the disorientation problem. These extensions represent the best example of coupling spatial and temporal cues that we are aware of.

In her "History List" extension, shown in Figure 9, Foss maintains an ordered list, rather than miniatures, of each notecard users have examined in the session.

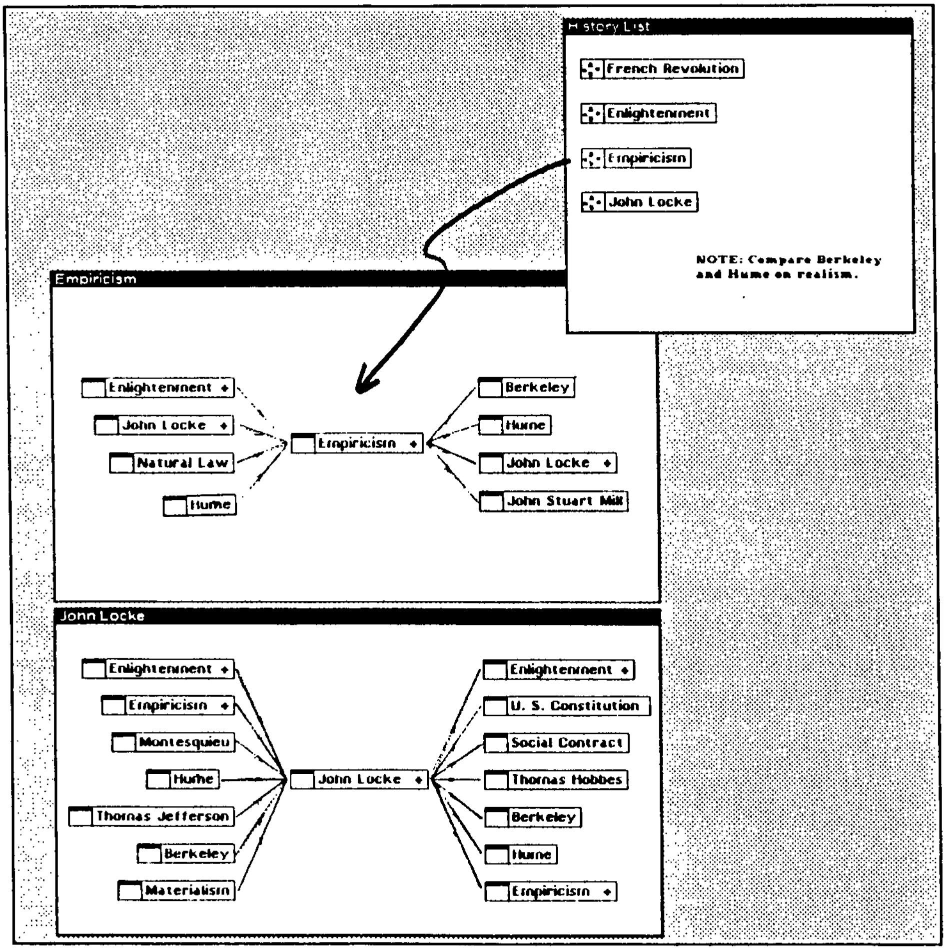

Fig. 9. A history list from Foss's NoteCards extensions.

Unlike users of HyperCard or EDS, NoteCards users can see this list on the screen while they are browsing through links and nodes. Users can annotate the History List by adding text at any place in the window. From the History List, users can select an item in the list and spawn a "minibrowser." This minibrowser shows the selected node in the center surrounded by all nodes that are connected to it. If users have already visited a node in the current session, the node in the minibrowser is marked with a plus (+) sign (one for each time visited).

EDS's Neighbors display is similar, but is not coupled with the "Timeline" feature, cannot be viewed in a separate window, and gives no indication of which neighboring nodes the user has already examined. In both NoteCards and EDS, links are unidirectional, so both Foss's minibrowser and the EDS Neighbors display indicate which nodes precede, and which follow, the focus node.

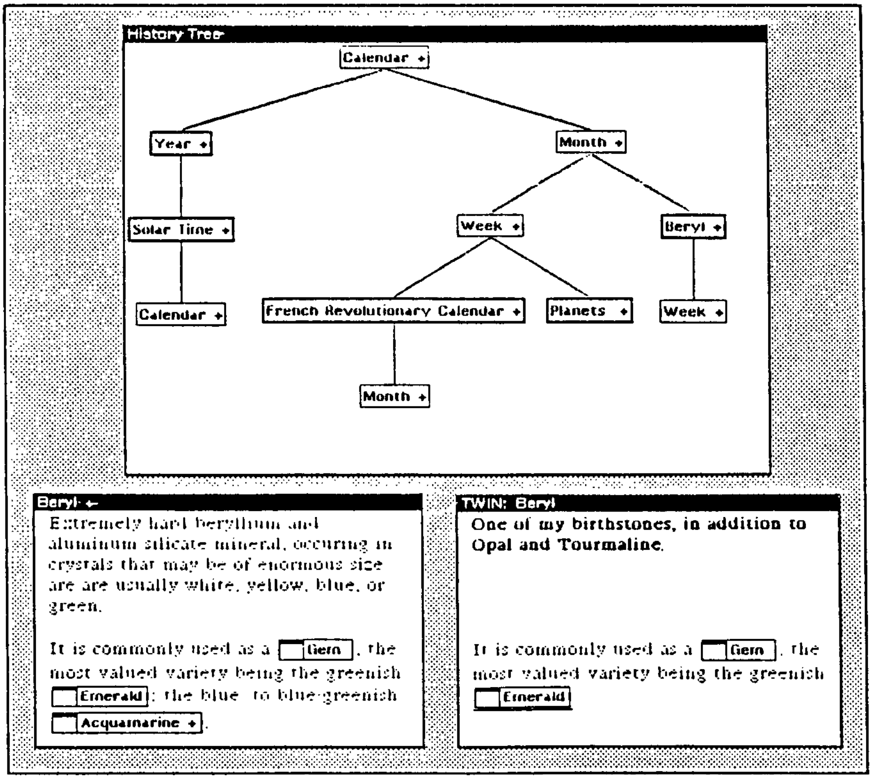

Fig. 10. A history tree from Foss's NoteCards extensions.

Another feature Foss added was a "History Tree," shown in Figure 10. Unlike the History List, the History Tree display is hierarchical rather than linear. The purpose of this display is to try to give users a sense of *how* they traversed a set of linked nodes. The Tree display attempts to diagram digressions and to point out nodes that have been examined multiple times. Like the History List, a version of the History Tree can be saved and annotated with text and graphics. While the idea of a historical diagram that contains more semantic information than a list is exciting, we believe that more work is necessary to design a diagram that is more intuitive to understand at a glance.

5. ALTERNATIVE APPROACHES FOR THE WEB VIEW

In upgrading the design of Intermedia, we sought those features of current systems that would help to achieve the goals we described above in Section 3. Features compatible with our goals also had to be compatible with our environment. Intermedia uniquely combines linking of potentially large documents (rather than small, limited "cards") with a multiwindow "desktop" environment, where multiple users share access to the same materials. What appealed to us most in the systems we examined was the coupling of the historical information, which we call the *path*, with the contextual information, which we call the *map*.

The key difference among our initial design proposals was the way in which each one defined and presented the layout of nodes in the web. Our idea was to create a global map for each web, assigning each document in the web a position in an arbitrarily large two-dimensional space. We felt initially that fixing a position for every document in the web would allow the user to develop a very precise sense both of location in the web and of the overall structure of the web, thereby solving, to a great extent, the issue of spatial context.

We believed that having fixed spatial positions allows users to improve recall by means of spatial imagery [2, 10] through which users can recognize clusters, as well as individual items. For example, certain groups of documents may form a distinctive pattern in a map. Referring to our scenario in Section 2, we may suppose that, over time, Susan may recognize that a group of documents all have to do with dental insurance. When looking for a particular document on this topic, Susan can scroll the map to that group and expect to find her target document in that general area.

Another potentially useful aspect of spatial layout is that it invites comparison. For example, assuming that closely related documents are placed together and unrelated documents are placed far apart, one can tell at a glance whether one pair of documents is more or less related than another pair.

Furthermore, spatial layout allows the user to develop a sense of the scope, or size, of the data space. Referring again to our scenario, it allows Susan to see how much material is in the web, and to thereby estimate how much of it she has already covered and how much effort it will take to read the rest of it.

Fundamentally, it seems that the most direct way to give users a sense of location, and thereby dispel their disorientation, is to form a map of the entire web space and show users their location in that space along with other indicators, such as how they got there and where they could go next. The use of this technique in Palenque reinforced our belief in the usefulness of this approach.

Several problems with spatial layout, however, dictated the rejection of all such proposals.

One problem is the sheer size of the layout. Our maps show the name of each document and an icon indicating the document's type. A web can contain any number of documents and, as stated earlier, in practice have been as large as 1000 documents and 2000 links. The large amount of space required for a complete view of such a web poses a problem for two reasons. First, it is hard for the user to move through such a large space; it requires much mouse movement, there are many patterns and clusters to remember, and so forth. Second, it becomes harder to get all relevant information into the view. That is, as the size of the entire map increases, it becomes more unlikely that all documents linked to a certain document will fit into the view at the same time.

These difficulties led us to consider various strategies concerned mainly with filtering or compacting the layout. Systems with data organized in a hierarchical structure can easily mitigate the size problem by collapsing and expanding branches of the hierarchy as desired. Intermedia links, however, form a network rather than a hierarchy. Our attempts to design a way to collapse and expand sections of our network were unsuccessful. We attempted to take advantage of the preexisting folder hierarchy to compact the map in a way similar to EDS's semiglobal view. However, we encountered the same problems mentioned earlier with regard to EDS; such a map lacks relevance to real user needs and is space intensive. Additionally, it seems awkward to have to manipulate Finder windows for the purpose of viewing the web.

Another strategy was based on a "fish-eye view" [15] of the network. As much of the map as could fit in a center section of the window would be displayed. This view would be dynamically shifted around so that the current (focus) document was always in this section of the window. Any documents linked to the focus, which were not displayed in this section, would be placed in a frame around the center section.

Although this approach fairly effectively resolved some of the problems of layout size, there were questions about whether frequently shifting a fairly narrow view around would ever allow the user to develop a useful cognitive image of the map. More fundamentally, the fish-eye approach failed to resolve the problems of actually defining the spatial layout. That is, how does the system decide how to position documents without requiring significant user intervention?

One possibility is to simply place each new document at a random position. This is essentially what was done in our original global map. It was unsatisfactory because of the hopeless mess caused by drawing the lines that indicate which documents are connected by links, and because a document's location conveyed no inherent meaning.

Another approach would be to let users determine the document's position. We rejected this method for two reasons. First, our goal, for reasons stated earlier, was that the Web View should be nonintrusive. Second, a location specified by a user is subjective, and other users sharing the same web might disagree with the placement decisions. We hoped instead to find a more objective placement strategy.

The most obvious strategy was to define the distance between two documents in a web as the smallest number of links one must follow to get from one of the documents to the other (documents that are not connected at all by links would be assigned some arbitrarily large distance). We called this the "minimal links placement" strategy. Given a table of the distances between each document pair, it is possible to assign every document a position in two-dimensional space such that the distances between them are closely, if not exactly, correlated to the calculated table [23]. Obviously, the precision of the positioning would have to be sacrificed to some degree to prevent the document icons from overlapping each other in the map, but this effect can be minimized given a large enough coordinate space to spread out the documents in. This approach would minimize the "spaghetti" caused by the random-placement method and would allow the kind of comparison between documents mentioned earlier. However it, too, encountered serious obstacles.

First, as new links are created and as new documents are added to the web, a map based on link structure has to be reorganized. Performing updates in real time is challenging. The time required to find the fewest number of links between all pairs of documents in a web, at least by a straightforward algorithm, has been shown to be proportionate to $D^2 + LD$, where D is the number of documents in the web, and L is the number of links in the web [22]. Simply stated, as documents are added to the web, the time required to update the map quickly grows into an unacceptable delay.

Second, the shifting of documents as the map is updated largely negates the advantages hoped for from a spatial layout. Spatial memory cannot be relied upon, and the pattern of a cluster of documents changes. And these changes are not predictable. That is, the user is not able to predict the effect that adding a link to the web will have on the map. Furthermore, given these problems, how can such a map prevent disorientation? If you leave home one morning, and return in the evening to find the buildings in your neighborhood in completely different locations, you'll feel pretty disoriented even if someone gives you a new map.

Third, it is not clear that minimal links is a reasonable basis for determining document distances. In a highly connected web, where most documents are linked to most other documents, the strategy becomes almost equivalent to random placement. And perhaps the distance between documents should be weighted based on other factors. For example, whereas the distance between two documents that share just one link should be defined as 1, the distance between documents that share three links might be defined as one-third. Another point to consider is that not all links have equal "strength of relationship." One link may be made to indicate a fairly peripheral connection between two documents, whereas another may indicate a crucial point of comparison.

These problems led us to consider a "semispatial" approach, in which only documents that a user decided were "key" would be assigned a fixed position. Other documents would "float" and be displayed next to any document which they were linked to when that document was the focus. Figure 11 shows how such an approach might work. In Figure 11a the user is at key document F and the nonkey, "ancillary" documents linked to it are positioned nearby. In Figure 11b the user follows a link to ancillary document g, and the map is unchanged. The user then follows a link from g to key document Q, and the map in Figure 11c is altered to position all documents linked to Q, including g, as close as possible to Q.

Ideally, this approach leads to a state in which relevant ancillary documents are clustered around key documents which the user has recently visited, and it is this self-organizing aspect that is most appealing. One difficulty encountered with semispatial approaches, as with all subjective strategies, was that it is vulnerable to disagreement between users. Different users may not agree on which documents should be the key documents or on their spatial relationships. Additionally, the large-scale shifting of documents again negates hoped-for benefits, such as spatial memory. Therefore, it was not clear that a semispatial map would significantly alleviate the problems of disorientation and cognitive overhead.

In the end, it was decided that a global map of a hypermedia network, at least in Intermedia's case, is not feasible. A serious problem is the sheer impracticality of a global view of a dynamic read-write multiuser hypertext network that is both stable and does not require significant user effort to organize. The physical space required for such a view, the time required to update a global map as the network changes, the likelihood of related documents being placed far apart in the layout,

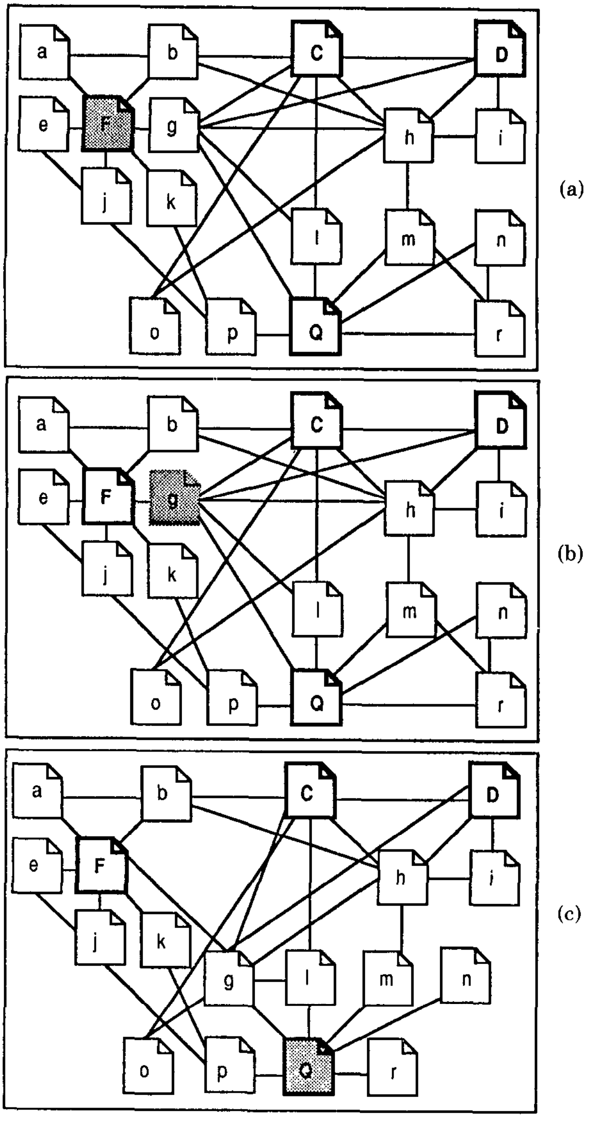

Fig. 11. Schematic semispatial view. Key documents are designated by uppercase letters. (a) User is at document F. (b) User follows link to document g. (c) User follows links to document Q.

and the lack of an objective basis for automatically positioning documents are all formidable obstacles.

Essentially, the link structure of the web has no inherent correlation with the user's concept of how documents are related. Global maps invite comparison between different documents and links, but making the proper comparisons necessarily requires the sort of subjective understanding that we find in the overview documents discussed earlier (Figure 4), and necessarily involves human intervention. Therefore, we decided to focus on local context, providing the user with tools for navigation rather than tools for comparison.

6. CHOSEN APPROACH

The Web View we developed has three major components: a *path*, a *map*, and a *scope line*, as illustrated in Figure 12. In addition, it provides three important features: the *dynamic updating* of the user's map, *link previewing* to give users more information about the target of a link before they follow it, and *shortcuts* to allow users to open documents and follow links from the Web View.

In Intermedia, the user opens a web by selecting a web document icon in the Finder and then the Open command from a menu (or by double clicking on the document icon). The system then creates the Web View window. Any documents opened or created from this point until the web is closed (by closing the Web View window) display the link information for that document in this web. A document may have different links in different webs. No more than one web may be open at a time.

6.1 The Web View's Features

The Web View's path is a list of the documents the user visits either by opening the document, by following a link, or by activating an already open document. Each of these actions is called a *path event*. The display of a path event consists of the name of the document, an icon indicating the type of document, an icon indicating the type of event (i.e., open, follow, or activate), and a timestamp indicating when the event occurred. Each user has a unique path for each web in the system. It is a simple linear list of the user's activity, comparable to the history list of the UNIX® command language interpreter c-shell (csh). The path supports browsing of past activity and the reopening (or activating) of documents in the path. Documents are added to the path even if they do not contain any blocks or links in the web. A user's path is saved when the web is closed and restored the next time the user opens that web.

The map is similar to the local tracking map in Intermedia's original implementation in that it displays all the documents linked to the current document and is updated as the current document changes. One change, however, is that the display is much more compact, as can be seen by comparing Figure 3 and Figure 12. The icons used to display the document type are smaller than they used to be, and they are laid out in alphabetical order in rows beneath the focus document, rather than in two long columns. Additionally, the layout algorithm considers the size of the window, so that as many documents as possible will fit into the window, as shown in Figures 13 and 14.

The scope line informs the user how many documents and links there are in the web. It therefore serves one of the functions of a global view in that it gives users a rudimentary sense of the size of the web.

The dynamic updating of the map is an important feature. Several systems provide maps of the link network that require an explicit request from the user to be updated, for example, the EDS Neighbors display or Foss's mini-browser. In these cases, therefore, the user must weigh the potential benefit of viewing

® UNIX is a trademark of AT&T Bell Laboratories.

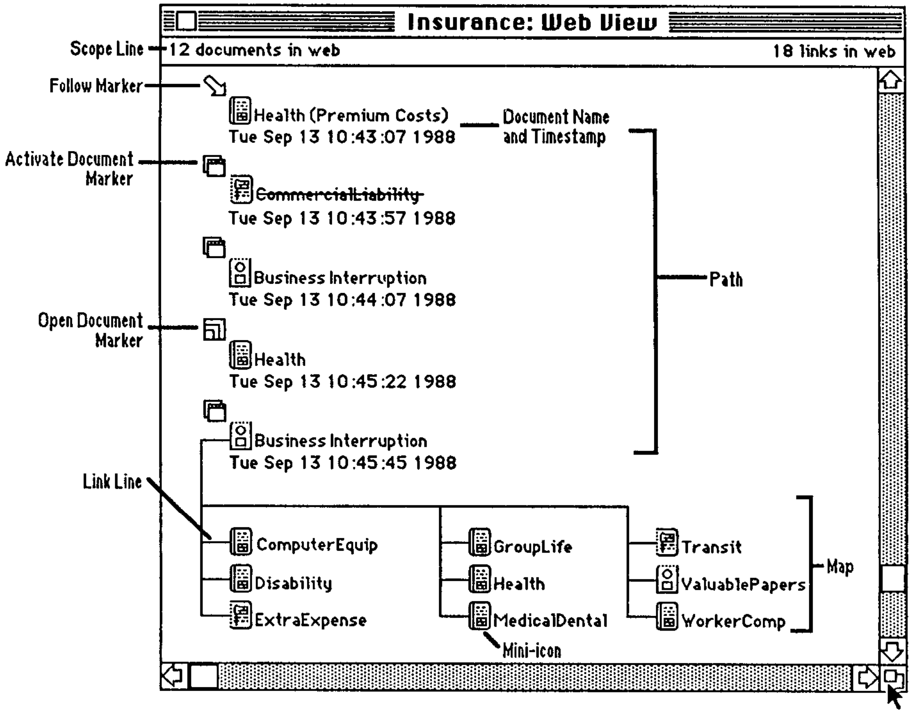

Fig. 12. Intermedia Web View.

the map versus the cost in time and effort in issuing the request to see the map. Dynamically updating the Web View map ensures that the information is always readily available.

Additionally, dynamic update is a prerequisite for another important feature of the map, link previewing. When the user selects a link marker in a document, the corresponding link line in the map is highlighted. Referring back to Figure 1, the user has selected a link marker that is connected to a block in the text document named *Bailee's*. This allows the user to see where a link leads without actually having to follow it.

Shortcuts allow any of the documents displayed in the path or map to be opened by selecting the document and choosing the Open menu command (or by double clicking on the document icon). In addition, by selecting a link line in the map and choosing the Follow menu command (or by double clicking on the link line), the user can follow a link from the current document to the other document associated with the selected link line.

6.2 Relevance of the Web View to the Scenario

Referring back to the scenario, we can see how the new Web View alleviates some of the problems Susan faced. One problem Susan encountered was that she wanted to avoid following links that would return her to a document she had already browsed. Link previewing clearly helps her in this regard. Susan can select a link marker and see which link line or lines are highlighted in the map.

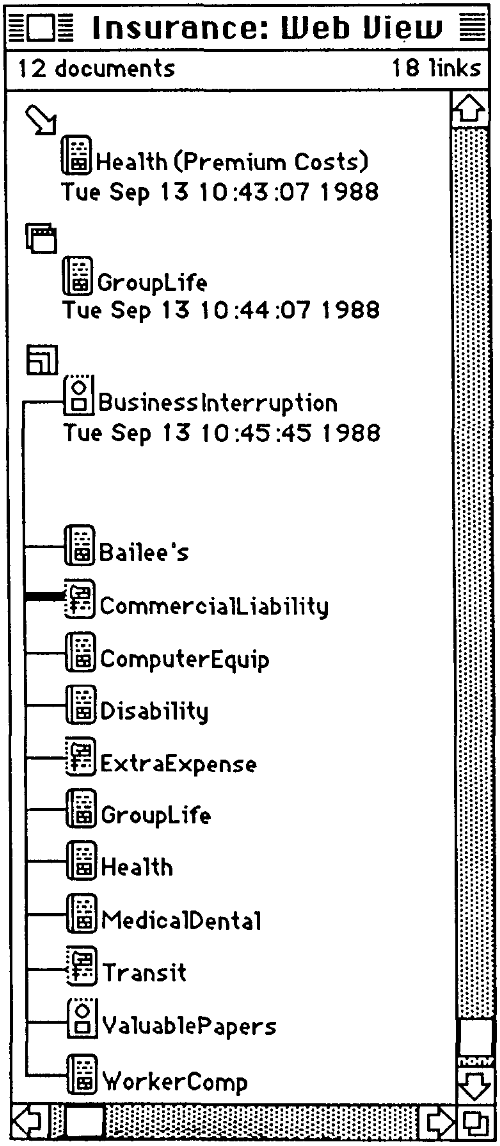

Fig. 13. Vertical layout.

If she can remember the names of the documents she has already browsed, she can immediately determine whether or not she should follow the link. If she cannot remember, she can activate the Web View window and scroll back through the path.

Another problem Susan had was trying to determine how much material is in the web. The scope line of the Web View tells her that, in this case, there are 12 documents and 18 links in the web. Assuming that it takes no more than a few minutes to scan through each document, and that there are a couple of hours to go before lunch, Susan should have plenty of time to scan the web.

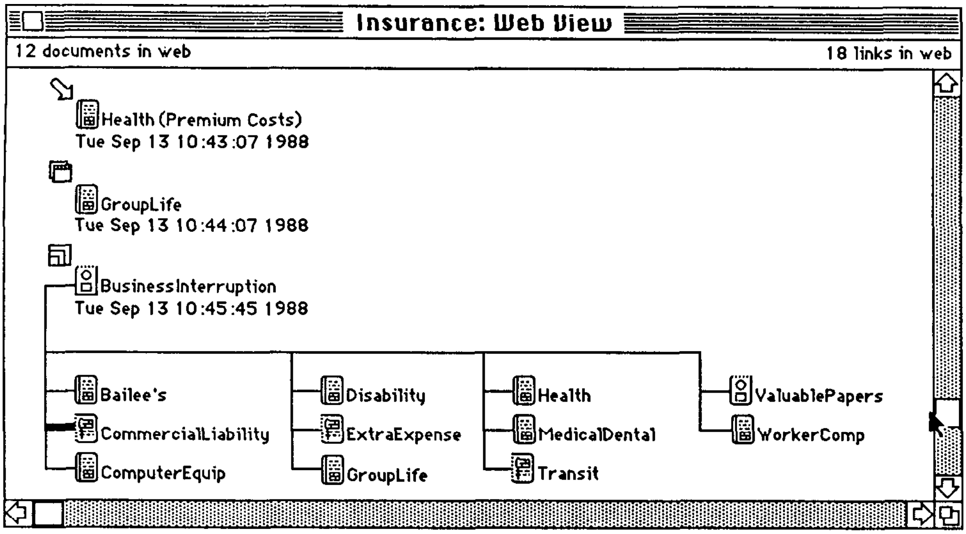

Fig. 14. Horizontal layout.

The scope line is also useful in a disjoint web. It is possible to open one document, follow all of its links and all the links in all the documents it opens, and so on until all connections have been fully explored, yet the user may not have examined the entire web. Other documents and links may also exist but be unconnected to the documents the user explored. This happened to one of the authors of this paper, who could have avoided the mistake if the scope line had existed at the time. The four or five documents read clearly did not amount to the twenty or so documents that the scope line would have reported.

Susan's third problem was trying to get back to the *GroupLife* document. In addition to the methods previously described for opening a document, such as activating the window or following links, the Web View shortcuts now provide another method. Susan can scroll back through the path until she finds an event for *GroupLife*, select it, and issue an Open command.

Another case in which the Web View shortcuts are useful is when a user is restarting a session. Suppose Susan restarts Intermedia after her one o'clock meeting and reopens the *Insurance* web. She can then scroll through the path looking for documents that she had opened previously and select all of them. Then she can issue the Open command and they will all be opened. Alternatively, she could just issue a Select All command, which selects all the documents in the path (except those that have been deleted), followed by the Open command.

Shortcuts work especially well in conjunction with link previewing. Suppose a user selects a link marker in a document but cannot see where it leads because the Web View window is obscured. The user can activate the Web View window to see where the link leads. With this information, the user may decide to follow the link. Rather than reactivating the original document to perform the Follow, the user can immediately perform the operation from the Web View window itself.

6.3 Relevance of the Web View to Disorientation and Cognitive Overhead

The redesigned Web View satisfies most of the goals stated in Section 3. The user is provided with spatial and temporal context in a flexible, nonintrusive manner. In doing so, the Web View helps to alleviate the problems of disorientation and cognitive overhead.

The path provides temporal context, allowing the user to see how the current state of the system was arrived at and providing a convenient mechanism for retrieving documents viewed earlier.

The map provides spatial context, allowing the user to answer the question of "where can I go from here." Interestingly, since links in Intermedia are always bidirectional, the map can also be thought of as showing "where can I get here from," or "who references me." Spatial context is also provided by the scope line, which, in a sense, allows the user to see the boundaries of the web space.

This information is provided in a nonintrusive manner. The map requires no user intervention to improve its appearance, and path events are also recorded and displayed by the system as the user goes about the primary task of developing and/or browsing the web. Similarly, link previewing occurs automatically, whenever any link marker is selected. In addition, the Web View is compact and flexible without being crowded.

By providing context, the Web View helps to address the problem of disorientation. As in the physical world, our sense of orientation in a hypermedia web is based on knowing where we are, not only in relation to some absolute frame of reference, but also in relation to other objects. Orientation is also based on knowing, or being able to make reasonable assumptions about, how to get from one object to another.

By being nonintrusive, the Web View helps to address the problem of cognitive overhead. For example, in a system that requires an explicit command to update the map, the user must interrupt a primary task for the secondary task of maintaining context. Furthermore, by putting as much of the map as possible in the window, we reduce the need for scrolling the Web View, again reducing cognitive overhead.

7. FUTURE WORK

There are still several ways in which the Web View could be improved. The user who is unfamiliar with the content of a web may find it difficult to find a good place to begin browsing. There is no way to determine which documents in the web are the "key" documents that might provide a broad overview of the contents of the web. Indeed, it can be difficult to determine if a document even contains any links in the web without actually opening the document. This problem might be alleviated if the Web View allowed the user to request information about the web such as "find the ten documents with the highest number of links," or "find the ten documents linked to the most other documents." Presumably, these documents would provide the best starting points for browsing a web. Also, if queries such as these can be formed, then more special-purpose queries could also be generated, such as "find all the graphics documents created in the past week that contain links with certain keywords."

A second issue is the level of detail that should be provided by the Web View. Links in Intermedia are not made between documents but between blocks (selection regions) within documents. To be conceptually complete, therefore, the Web View must provide this level of information. Indeed, the provision of this information was included in the design of the Web View but was left out of the implementation. The user was to be able to select any document in the Web View and see its blocks, listed by name, in a pop-up menu. The menu was to be hierarchical, with a second level displaying the blocks linked to each of the document's blocks. Although unimplemented, it seems possible that such functionality would help make the Web View more effective in reducing disorientation by being more precise in the information it provides.

As another issue, part of the power of the Web View is derived from the ability of the user to open documents from it. It seems possible, then, that by adding other Finder-like capabilities, we could increase the Web View's power further. Allowing a Close command seems natural and symmetric with the Open command. Renaming and deleting documents also seem possible. On the other hand, it can be argued that the Web View is not a Finder, and that adding such features would create a confusing overlap in responsibilities between the two facilities.

Finally, a major issue we have considered concerns sharing paths among users. It might be useful, for instance, for a professor to "blaze a trail" through the major documents of a corpus and give it to students so they could "replay" it to get a brief overview of the material. Or perhaps a user who refers to a small set of documents frequently might want to collect them all into a path so they could be quickly found and opened.

Another interesting aspect of paths is their "linearizing" function. That is, a path defines a specific sequence of nodes of the hypertext network. It may therefore be seen as making a "linear document" out of pieces of the network. In fact, this view of paths dates back to Bush's Memex [3, 4]. In Intermedia, however, which has relatively large nodes that often form complete documents in their own right, paths are less useful in this regard, and therefore the implementation of this sort of feature is less imperative.

If paths are to be put to such a purpose, additional features would be necessary. For example, users may want to turn path collection on and off, or may want to cut, copy, and paste portions of a path. Also, it would be helpful if some features were added to the system itself. For example, our paths record the opening and activating of documents and the following of links. But events such as scrolling a document, selecting a menu command, clicking the mouse or pressing a key are also important when trying to replay a user's activity, so the system should make it possible to record these events. As another example, the contents of a document may have changed or been deleted since a path was created. It would be helpful, then, if the system could store different versions of a document so that the path could be guaranteed to always bring the user to the same version of a document that the creator of the path used.

Essentially, sharing and replaying paths is a major research and design effort in its own right. Because it does not directly deal with the issues of context and orientation, our project did not address sharing and replaying paths (see [24] and [29]).

8. CONCLUSION

The problem of user disorientation in hypermedia networks has been noted by Conklin and others. It has been reported by participants in Intermedia field trials, is borne out by our personal experience, and may also be familiar to readers who have used HyperCard or other hypermedia systems. The approach we have implemented provides much of the information necessary to reduce the confusion and does so in a nonintrusive manner, requiring essentially no extra effort by users.

Intermedia's new Web View enhances system predictability by providing at least two clues of where a link is headed: the type of the document and its name. The Web View is also useful because it allows users to focus on manipulating the material in the web rather than on manipulating the web itself. The path information displayed in the view is automatically collected by the system, the map requires no action or information from users beyond what is necessary to make a link, and link previewing requires no action or information from users beyond what is necessary to select a link marker. In addition, the view provides shortcuts so that users attempting to take advantage of its information are not inconvenienced in doing so. Information that is problematic to get is often never gotten at all.

Like networks of roads, hypermedia networks need road signs: clues to what lies ahead, how far ahead it lies, and what type of material or service it will offer. And there must be a trail, a way of describing how you got to your present location. And once you have found a place, it should be easier to get there the next time. Intermedia's Web View provides just such a hypermedia road map.

REFERENCES

1. BEEMAN, W., ANDERSON, K., BADER, G., LARKIN, J., MCCLARD, A., MCQUILLAN, P., AND SHIELDS, M. Hypertext and pluralism: From lineal to non-lineal thinking. In *Hypertext '87 Papers* (Chapel Hill, N.C., Nov. 1987). University of North Carolina, Chapel Hill, 1987, pp. 67–88.
2. BOLT, R. *Spatial Data Management.* LCN 78-78256, MIT, Cambridge, Mass., 1979.
3. BUSH, V. As we may think. *Atlantic Monthly* (July 1945), 101–108.
4. BUSH, V. Memex revisited. In *Science Is Not Enough.* William Morrow, New York, 1967, pp. 75–101.
5. CATANO, J. Poetry and computers: Experimenting with communal text. *Comput. Hum. 13* (1979), 269–275.
6. COLLIER, G. Thoth-II: Hypertext with explicit semantics. In *Hypertext '87 Papers* (Chapel Hill, N.C., Nov. 1987). University of North Carolina, Chapel Hill, 1987, pp. 269–287.
7. CONKLIN, J. Hypertext: An introduction and survey. *IEEE Comput. 20*, 9 (1987), 17–41.
8. CONKLIN, J., AND BEGEMAN, M. gIBIS: A hypertext tool for team design deliberation. *ACM Trans. Off. Inf. Syst. 6*, 4 (Oct. 1988), 303–331.
9. DELISLE, N., AND SCHWARTZ, M. Neptune: A hypertext system for CAD applications. In *Proceedings of the International Conference on the Management of Data* (Washington, D.C., May 28–30, 1986). ACM, New York, 1986, pp. 132–142.
10. DONELSON, W. Spatial management of information. *Comput. Graph.* 12, 3 (1978), 203–209.
11. FEINER, S. Interactive documents. In *Design in the Information Environment.* Southern Illinois University Press, Carbondale, Ill., 1985, pp. 118–132.
12. FEINER, S. Seeing the forest for the trees: Hierarchical display of hypertext structure. In *Proceedings of the Conference on Office Information Systems* (Palo Alto, Calif., Mar. 23–25, 1988). ACM, New York, 1988, pp. 205–212.

13. FEINER, S., NAGY, S., AND VAN DAM, A. An experimental system for creating and presenting interactive graphical documents. *ACM Trans. Graph. 1*, 1 (Jan. 1982), 59–77.
14. FOSS, C. Effective browsing in hypertext systems. In *RIAO '88 Conference Proceedings* (Cambridge, Mass., Mar. 1988). MIT, Cambridge, Mass., 1988, pp. 82–98.
15. FURNAS, G. W. Generalized fisheye views. In *Proceedings of CHI '86 Human Factors in Computing Systems* (Boston, Mass., Apr. 13–19, 1986). ACM, New York, 1986, pp. 16–23.
16. HALASZ, F. Reflections on Notecards: Seven issues for the next generation of hypermedia systems. *Commun. ACM 31*, 7 (July 1988), 836–851.
17. HALASZ, F., MORAN, T., AND TRIGG, R. NoteCards in a nutshell. In *CHI/GI '87 Human Factors in Computing Systems and Graphics Interface* (Toronto, Ontario, Apr. 5–9, 1987). ACM, New York, 1987, pp. 45–52.
18. HAMMOND, N., AND ALLINSON, L. Travels around a learning support environment: Rambling, orienteering or touring? In *Proceedings of CHI '88: Human Factors in Computing Systems* (Washington, D.C., May 15–19, 1988). ACM, New York, 1988, pp. 269–273.
19. HILL, B. *Guide: Hypertext for the Macintosh.* Owl International, Inc., Bellevue, Wash., 1986, pp. 25–39.
20. KAEHLER, C. *HyperCard Power: Techniques and Scripts.* Addison-Wesley, Reading, Mass., 1988, pp. 18–19.
21. LANDOW, G. Hypertext in literary education, criticism, and scholarship. *Comput. Hum. 23*, 2 (1989), 15–30.
22. SEDGEWICK, R. *Algorithms.* Addison-Wesley, Reading, Mass., 1988, p. 477.
23. SPENCER, R. *Similarity mapping. Byte 11*, 8 (1986), 85–92.
24. TRIGG, R. Guided tours and tabletops: Tools for communicating in a hypertext environment. *ACM Trans. Off. Inf. Syst. 6*, 4 (Oct. 1988), 398–414.
25. WALKER, J. Document Examiner: Delivery interface for hypertext documents. In *Hypertext '87 Papers* (Chapel Hill, N.C., Nov. 1987). University of North Carolina, Chapel Hill, 1987, pp. 307–323.
26. WILSON, K. Palenque: An interactive multimedia optical disc prototype for children. Working Paper No. 2, Bank Street College of Education, New York, March, 1987.
27. YANKELOVICH, N., MEYROWITZ, N., AND VAN DAM, A. Reading and writing the electronic book. *IEEE Computer 18*, 10 (1985), 15–30.
28. YANKELOVICH, N., HAAN, B., MEYROWITZ, N., AND DRUCKER, S. Intermedia: The concept and the construction of a seamless information environment. *IEEE Computer 21*, 1 (1988), 81–96.
29. ZELLWEGER, P. Active paths through multimedia documents. In *Proceedings of the International Conference on Electronic Publishing, Document Manipulation, and Typography* (Nice, Apr. 1988). Cambridge University Press, Cambridge, England, 1988, pp. 19–34.