12-31-1994

A New File Access Paradigm for Extending a User's Information Base

Jane Fritz
University of New Brunswick

Ian Benest

A NEW FILE ACCESS PARADIGM FOR EXTENDING A USER'S INFORMATION BASE

Jane M. Fritz
Faculty of Computer Science
University of New Brunswick

Ian D. Benest
Department of Computer Science
University of York

ABSTRACT

Information systems and decision support systems have traditionally provided quantitative interpretations of quantified data. Strictly speaking, information systems address structured problems and decision support systems support semi-structured and even unstructured problems, but both classes of information provision restrict their domain to data that has been transformed into database entries. Intrinsic meaning that may have been attached to the data in document form is lost. Traditionally, they ignore much potentially valuable information that resides in non-quantifiable form, such as memos, reports, pictures, and drawings. They also ignore those facets of decision making that draw on a heuristic approach rather than an analytical one. Information systems continue to exist within the artificial constraints of monolithic computer applications, restricting the information space to data that is "attached" to that application. In this paper we propose a file management model that makes use of hypertext-type techniques to provide a user with the ability to store meanings and relationships as well as the files themselves. We submit that the inclusion of this unifying hypertext facility can provide a common interface that will extend the accessibility and utility of the user's information base. This open hypertext environment can provide the mechanism by which we can bring nonquantified information into the domain of computer-based support for decision making. A prototype implementation is discussed, as well as observations resulting from initial studies.

1. BACKGROUND

Decision making has been identified by Simon (1960) as consisting of three distinct phases: intelligence, design, and choice. Intelligence refers to the monitoring of the environment for problems, challenges, and opportunities; design refers to the generation of solutions to identified problems or opportunities; choice is the actual choosing of the "best" solution. Existing computer support for decision making concentrates on the choosing. Additionally, it is recognized that humans rely at least as much on intuition as analytical reasoning in all phases of the decision making process (Dreyfus and Dreyfus 1988; Dreyfus 1992; March 1991; Winograd and Flores 1987; Dawson 1993; Young 1989), and yet computers support and indeed mimic analytical reasoning, not intuition. One of the fundamental elements of creativity is exploring the problem space created by the decision maker's interpretation of interrelations between nodes of information (Sternberg 1987; Boden 1991; Rettig 1992). A person's interpretations, definitions of interrelationships, and intuitive reasoning based on context and experience are notoriously difficult to automate, and so are notably ignored or downplayed in computer support. Putting structure on ideas is key to understanding, and yet as the understanding unfolds and deepens, the structure changes. Aspects of decision making, particularly the intelligence and design phases, rely heavily on these processes of categorizing and recategorizing, associating and reassociating; still aids for these vital processes are remarkably primitive in our everyday computing environments. Reliance on current computer-based support encourages the user to develop a convergent, quantitative approach to problem solving at the expense of a possible divergent, heuristic approach. To exploit fully the computer's potential to support human information processing, we must allow users to define associations and interrelations among files and be able to navigate their computer-based information space accordingly.

The concept that the entire body of information accessible from a user's workstation can represent the user's knowledge space is not supported by existing file management tools. In fact, this knowledge space contains nodes of information that may be semantically interrelated in many ways. The semantic connections exist in the mind of the user. For example, several files of diverse file types might be related in that they all describe a particular project. One or more of those files or file segments, along with others, might comprise the spreadsheets and templates needed to assemble an annual year-end document. The provision of an interface that allows users to navigate through their information spaces according to user-defined interrelations would facilitate the removal of artificial boundaries among existing information. The boundaries are established by the insularity of existing applications. While this insularity is being attacked by such features as Microsoft Windows' OLE and DDE, the inter-application connections resulting from OLE and DDE are still embedded within one application. Our model provides a common interface that in a sense removes the application boundaries.

Hypertext has been identified as having potential in many business applications (Mahapatra and Courtney 1992; Bieber and Kimbrough 1992). Most of these suggestions involve building new hypertext applications which enforce the notion of a hypertext as an electronic document. Our concept views an end user's computing environment itself as a hypertext. We have implemented a prototype with which we are investigating the effectiveness of open hypertext techniques for qualitative computer-based decision support. By open hypertext we mean the ability to define and navigate links from within any user application, using an interface common to all applications. In addition, links may be defined between nodes that are "attached" to separate applications. In effect, all of a user's computer-accessible files (and segments thereof) can participate as nodes in a user-defined hypertext. This contrasts with existing hypertext systems, which are typically monolithic in nature. Open hypertext is defined fully in Davis et al. (1992). The term "qualitative decision making" is meant to encompass all phases of decision making, or general problem solving, for which qualitative approaches are appropriate. A qualitative approach to decision making may be referred to as intuitive or heuristic. Integrated environments exist to overcome this shortcoming, but the interrelations in an integrated environment are usually predefined: data flow diagrams connected to data dictionary entries, report templates connected to predefined queries, etc. The environment is integrated to support a clearly defined task. The integrated paradigm does not extend to the whole of a user's potential information space. The integrated computer environment resulting from our open hypertext facility, in fact, extends the definition of information systems and decision support systems.

2. HYPERTEXT REVIEW

Hypertext is loosely defined as nodes of information that are linked according to some understood relationships. Subsequently, the nodes may be accessed in a non-linear fashion through the resulting link structures. The broader definition of hypertext, sometimes referred to as hypermedia, includes nodes that may contain text, pictures, video, sound, etc. The hypertext itself is often thought of as the navigable entity that results from the definition of nodes and links. Hypertext systems provide developers with the framework within which one may develop a hypertext for a particular use, such as a museum guide or a document development environment. These are monolithic, single-purpose hypertexts. The use of hypertext systems, or at least hypertext techniques, for general-purpose information retrieval has been studied from many vantage points, with the obvious goal of providing optimum access to required information. These studies concentrate on the problem of indexing and forming links for optimum retrieval results in settings where the user does not necessarily know what information might be available *and* the system does not have a predefined knowledge of the user's context. In other words, the indexing and linking process has to meet unknown needs.

The information space represented by a user's personal computer environment is a very different situation. In a business setting, or in any personal setting for that matter, relevant indexing will not necessarily be best determined by word frequencies. It might be the case that department names, project names, position titles, or even document type (memo, form, etc.) are most useful as indices, but are not found per se in a file. Also, many file types do not contain searchable text and so *must* be manually indexed. An open hypertext facility can include a simple method for user definition of node attributes and node-to-node links. It must also provide the flexibility to alter these definitions in an easy manner as an expanded understanding of a mental model unfolds. These are activities which cannot be determined by automation, but should not be avoided because of that. The results of these actions, user-defined hypertexts of personally organized data and information, transforms the users' computer-accessible files into a far more valuable resource for supporting the full range of decision making processes.

Some of the decision making processes we are concerned with are being addressed by the use of specialized hypertext systems and idea processors. We refer in particular to knowledge structuring tasks. For example, hypertext tools like NoteCards (Halasz 1988) and Aquanet (Marshall et al. 1991) have been used to guide collaborative efforts in organizing information into interpretive structures. In these cases, in order to provide a hypertext environment to support the anticipated evolution of structure, each project had to be mounted within the chosen hypertext system, rather than making use of the user's everyday computing environment. So, although the value of hypertext techniques was recognized for this important type of work, the techniques were not generically available. Marshall and Rogers (1992) make some pertinent observations as a result of their work with an Aquanet project. They found that users rarely take advantage of the ability to define complex relationships. It is more natural for them graphically to group nodes they understand as being related. It became clear that it is up to the users to manage their own interpretive structures; this cannot be automated. In our opinion this is a very important point, since many people believe that if it cannot be automated, it is not a worthwhile tool. We find this a narrow point of view, since it discounts the strengths of human information processing, including the understanding of context and experience, which have been found to be impossible to encode with any degree of success.

Within the boundaries of database management systems, it is clear that the stored data is not useful until some structure has been applied through joins, intersections, or queries generated by someone who brings some previous knowledge to the request and who adds a personal interpretation to the result. Object-oriented databases and intelligent databases (Parsaye et al. 1991) attempt to encode some meaning to data in the form of known rules, relationships, or methods. We submit that an analogy should be extended to a user's information base as a whole. Our design superimposes user-defined, hypertext-type link structures on a user's information base to meet this challenge.

3. OPEN HYPERTEXT MODEL

A new file access paradigm is required. Open hypertext features can provide that paradigm. Open, or cross-application, hypertext architectures have been designed (Kacmar and Leggett 1991), and instances of the concept have been implemented (Yankelovich et al. 1988; Haan et al. 1992 , Davis et al. 1992). Brown University's IRIS system (Haan et al. 1992) was the first one to capture the spirit of open hypertext, although it was forced to develop its own set of applications so that it could embed their hypertext features within each application. Its uses, which were extensive, were oriented toward developing enhanced learning environments. Sadly, the IRIS project, one of the leaders in the field, has had its development terminated because the Apple operating system on which they had based their applications became obsolete. Microcosm (Davis et al. 1992), developed at the University of Southampton, is very close in its system specifications to those identified for our work. It provides a strong information retrieval engine for search, and for what Microcosm calls computed links. It does not, however, provide graphical browsers of any kind, so that the relationships determined by common content or direct links are not given a visual representation. Nor can any hierarchical or other spatial relationship be visually represented. Reported studies into interfaces for a diversity of user groups, including hypertext users (Marshall et al. 1992; Nielsen 1990), programmers (Shu 1988), and users of information systems (Booth 1989) indicate that visual representations are important to understanding, and that the most effective visualization is dependent on the individual. Our initial experimental results reinforce this finding.

Our hypertext system differs from other open systems reported in the literature in that we provide end users with the capability to define and manage their own link structures for their own reasons. Most existing hypertext systems concentrate on providing tools for developers to design prestructured hypertext environments on behalf of end users. The resulting hypertext environments thus developed are more controlled in that assumptions on expected use are made and biases built in. Also, because of this predetermination. (and restriction) of use, certain implementation and navigation problems can be avoided. In our case, in providing end users with the ability to turn their information space into a self-determined, self-regulated hypertext-like entity, open-ended problems have to be identified and overcome. Some problems of implementation will only be overcome by embedding functionality such as file attribute capture and storage in the resident operating system. Other problems will be overcome by looking at new approaches to interface design. Our hypertext-like extensions to file management provide many features which will allow a user subsequently to develop custom hypertexts for other users, but that is not its primary goal. The main objective of these features is not to facilitate the concept of an electronic document so much as to expand the accessibility of all computer-accessible information on a user's electronic desk.

Figure 1 illustrates the overall concept of our design. We prefer not to think of it so much as a system as something that would ideally be incorporated seamlessly into every application, similar to cut and paste operations. We will refer to it as the Link Facility. For the purposes of our study, a user should be able to access a number of hypertext-type features from within any application that supports a file. We have identified the following features to be minimal requirements, incorporating some which have been positively reviewed in reports on use (Halasz 1988; Conklin and Begeman 1988; Marshall and Rogers 1992) and some whose omission has been lamented (Halasz 1988; Haan et al. 1992):

1. user-defined direct links;
2. keyword (and other attribute) assignments for information nodes;
3. user-defined link structures, dynamically configurable link structures;
4. filtering by query (search by query);
5. provision for nodes being defined as composites (e.g., an aggregate or a previously defined link structure);
6. annotation for any information node;
7. bookmarks;
8. provision for user-*redefinition* of stored keywords and other attributes;
9. provision for user-defined link types;
10. active graphical displays of link structures (search by browsing); and,
11. a maintenance capability.

In addition, we include the availability of generic templates for the recording of thoughts, design rationales, etc.

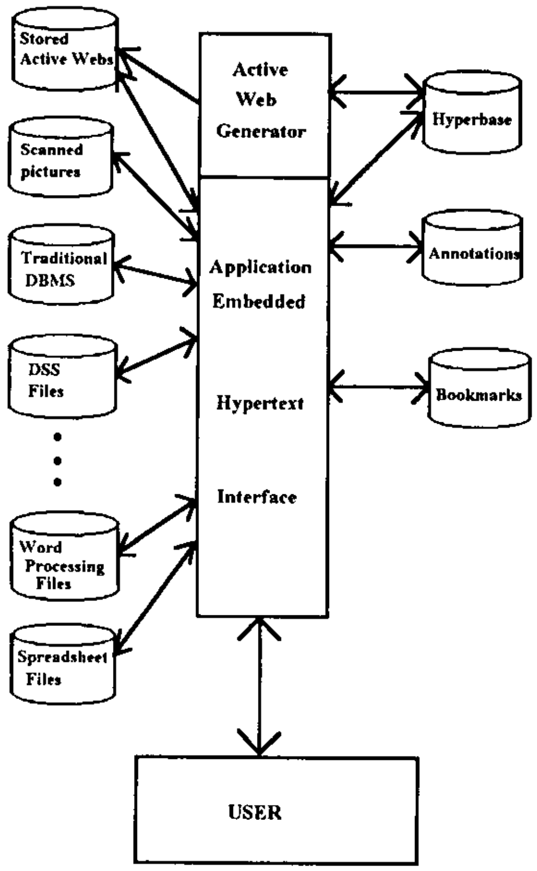


Figure 1. System Architecture

4. OUR LINK FACILITY PROTOTYPE

We have designed and implemented an open (or cross-application) hypertext prototype that runs in Microsoft Windows 3.1 applications through menu items embedded in participating applications. The Link Facility provides these hypertext capabilities directly within Word for Windows, Excel for Windows, and all ToolBook applications. Other application files can participate at the file level through a desktop icon. Ideally, of course, *any* application would have the facility embedded. Windows has been chosen over X Windows and OS/2 as the prototype computer environment because of its widespread use in the workplace within our test region. The details of the data model on which the Link Facility prototype is based are beyond the scope of this paper, but in substance it can be thought of as following the Dexter Reference Model (Halasz and Schwartz 1990). The data manipulation language, which is used and controlled by end users, is manifested as menu items and graphical manipulations. Because the prototype is implemented in Windows applications such as an Excel database and ToolBook graphical applications, the integrity of the link structures and node contents is maintained by Windows itself. If one user has opened a Windows file in write mode, then Windows will allow a concurrent user to access the same file only in read mode.

The main features are described in this section, after which strengths and limitations are discussed. Figure 2 shows the menu items available through participating applications and through the Link Facility desktop accessory.

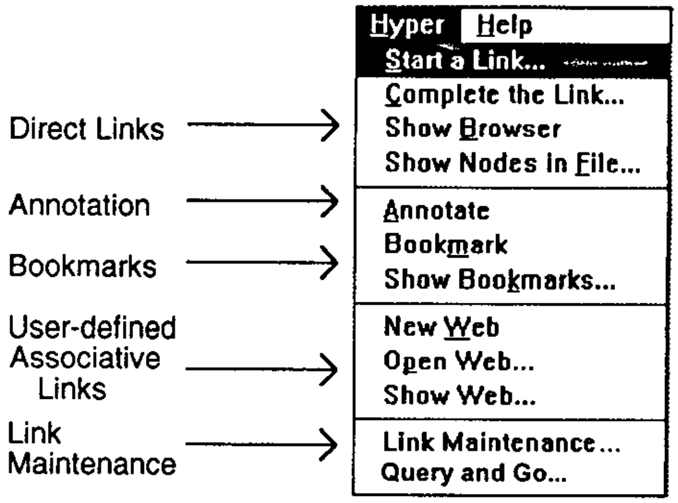


Figure 2. Menu Items Common to All Applications

4.1 Feature Descriptions

Browsers. Every file has a browser associated with it that graphically displays all direct links associated with all nodes in the file. Direct links are defined by the *StartLink* and *CompleteLink* actions. A browser is displayed by invoking the *Show Browser* menu item. The browser comes up in a separate window, as shown in Figure 3. The user may traverse the linked files simply by clicking on the graphical nodes.

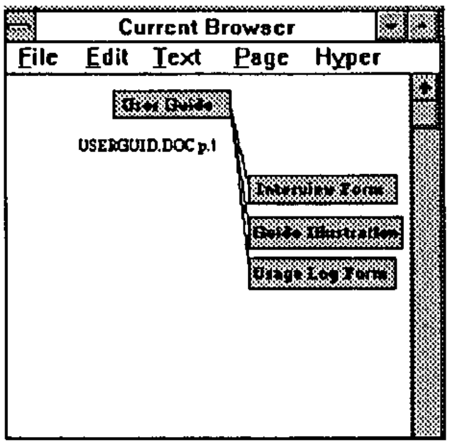
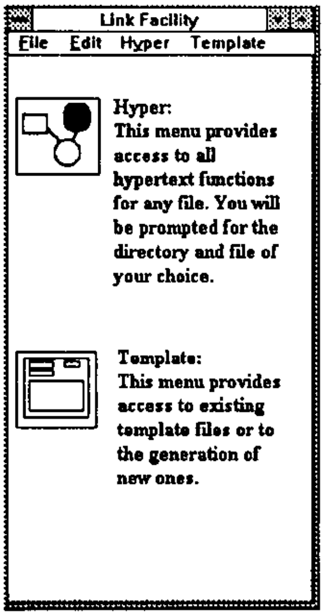


Figure 3. A Browser

Link Facility. The Link Facility is a desktop accessory that provides access to hypertext features for files that currently do not have embedded hypertext menu items. It also provides access to the Template feature, described below. Figure 4 shows the Link Facility interface when its desktop icon is maximized.

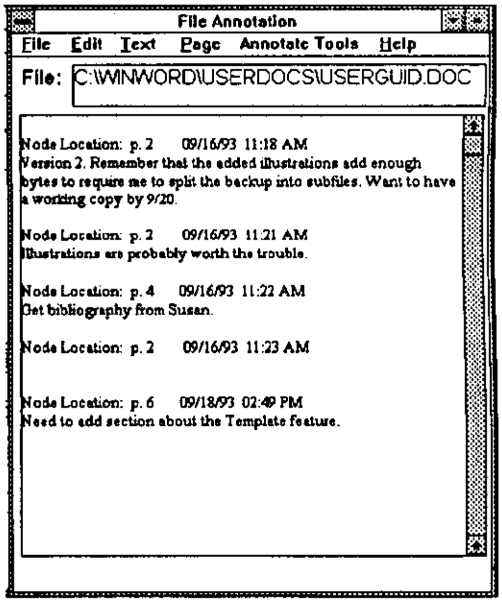


Hyper:
This menu provides access to all hypertext functions for any file. You will be prompted for the directory and file of your choice.

Template:
This menu provides access to existing template files or to the generation of new ones.

Figure 4. The Link Facility Interface

Annotation. Annotation can be invoked from within the active application, or from the Link Facility for a nonparticipating file. A comment "file" is created, automatically linked to the file to be annotated. It can be retrieved by the same menu item and expanded or changed at any time. Figure 5 shows a file annotation.



File: C:\WINWORD\USERDOCS\USERGUID.DOC

Node Location: p. 2 09/16/93 11:18 AM
Version 2. Remember that the added illustrations add enough bytes to require me to split the backup into subfiles. Want to have a working copy by 9/20.

Node Location: p. 2 09/16/93 11:21 AM
Illustrations are probably worth the trouble.

Node Location: p. 4 09/16/93 11:22 AM
Get bibliography from Susan.

Node Location: p. 2 09/16/93 11:23 AM

Node Location: p. 6 09/18/93 02:49 PM
Need to add section about the Template feature.

Figure 5. A File Annotation

Webs. Webs are graphical representations (or maps) of relations among nodes that are created by the user as a result of query action. They are similar to browsers in that the active link buttons come up in a window separate from the active file, but they are not dynamically generated by predefined links; rather, they are developed by virtue of the user selecting related nodes by querying on attribute criteria, then specifying which nodes to add to a web. The arrangement of nodes and connecting lines on the web, which defines the relationships among the nodes, is strictly a drag-and-drop operation provided by the software. Annotation buttons may be included to annotate links. Webs may participate as nodes in other webs. A user can display all stored webs for which the active file is a member by invoking the *Show Webs* menu item.

Notice that some nodes are not connected by lines (links). Such visual links are included by a user only to clarify their own understanding of relationships. Another user might have created the visualization of these interrelationships quite differently.

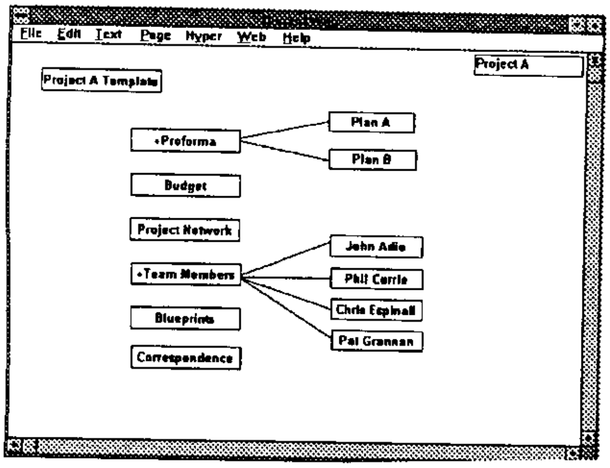
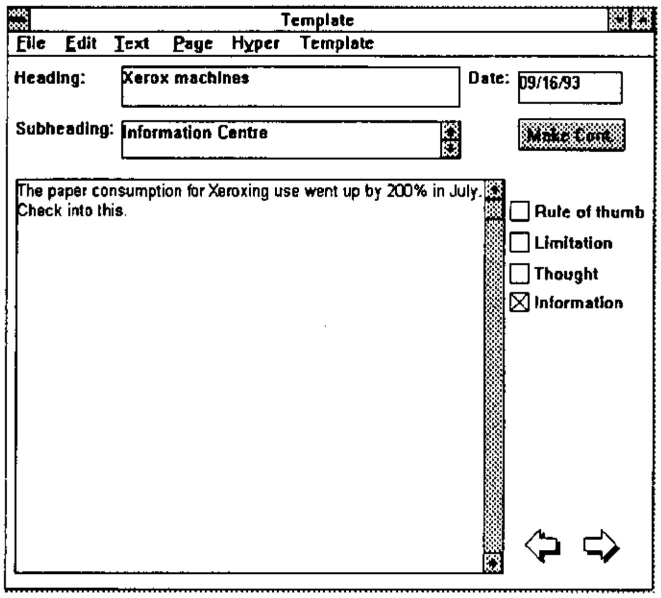


Figure 6. A Web

Templates. A template is a "pad" of blank forms that sit on a desktop for the user's convenience. Templates might be used to keep track of new ideas, keep client profiles, project histories, or decision rationales. As can be seen from Figure 7, the default form design includes the capturing of a "heading" category and within that an optional "subheading." These values are strictly to help the user organize ideas. Furthermore, the number of category fields and their names can be customized. For example, a user might want three category fields called "product," "customer name," and "customer address." The button labeled "Make Cont." generates a new screen on which to add additional comments for the current heading and subheading. The forward and back arrows will page the user through the existing collection of screens stored for the template of interest. Optionally, a screen may be checked as being any combination of "rule of thumb," "limitation," "thought," or "information." Notice that the hypertext features are available through the inclusion of the common *Hyper* menu, so that any of these templates may participate in a browser or a web. Printed reports of template information may be generated through a menu item. Templates are retrieved or created through the desktop Link Facility.

Attribute Redefinition. Attributes can be globally edited through the *Link Maintenance* menu item. This feature provides support for the cognitive tasks of recategorizing and restructuring as a problem or situation is better understood. Attribute values may be reassigned to (or deleted from) all or some of the existing stored nodes under the user's direction. Restructuring might be accommodated by subdividing a previous assignation into two or more separate value assignments. For example, all nodes previously given the keyword "budget" might be reassigned to one of "operating budget" or "capital requests." Other *Link Maintenance* functions include setting up global attribute choices and deleting dead links. Attributes can be used directly to retrieve a file by invoking the *Query and Go* menu item.



Heading: Xerox machines
Date: 09/16/93
Subheading: Information Centre

The paper consumption for Xeroxing use went up by 200% in July.
Check into this.

☐ Rule of thumb
☐ Limitation
☐ Thought
☒ Information

Figure 7. A Template

4.2 Functionality

Each file accessible from a user's workstation may have a browser and a collection of user-defined webs. A web can be retrieved by invoking *Open Web* and choosing from the entire collection, from a *Query and Go* using assigned attributes, or from *Show Webs* from the active file. Looking at Figure 8, we see a schematic diagram of a web's interrelationship with files and other webs. The central web in the diagram (Project A) is shown in full in Figure 6. If the user clicks on the node labeled "John Adie," an active window displaying that file (emplyjfa.doc) appears. Note that, from this file, the user can bring up any other associated web, from which another node can be opened in a new window. The new file will have its own browser and set of webs. These artifacts define the user's hypertext and serve as the navigation mechanism.

Our facility provides the following distinctive characteristics:

- The information stored in the resulting environment is conceptually globally-accessible rather than application-based.

- Node attributes may be locally redefined, globally redefined, or globally subdivided as the relevance of information is better understood.

- Browsers may be dynamically generated from user-defined direct links between nodes. They may also be user-configured as a result of attribute query and/or ad hoc inclusion of desired files (called webs).

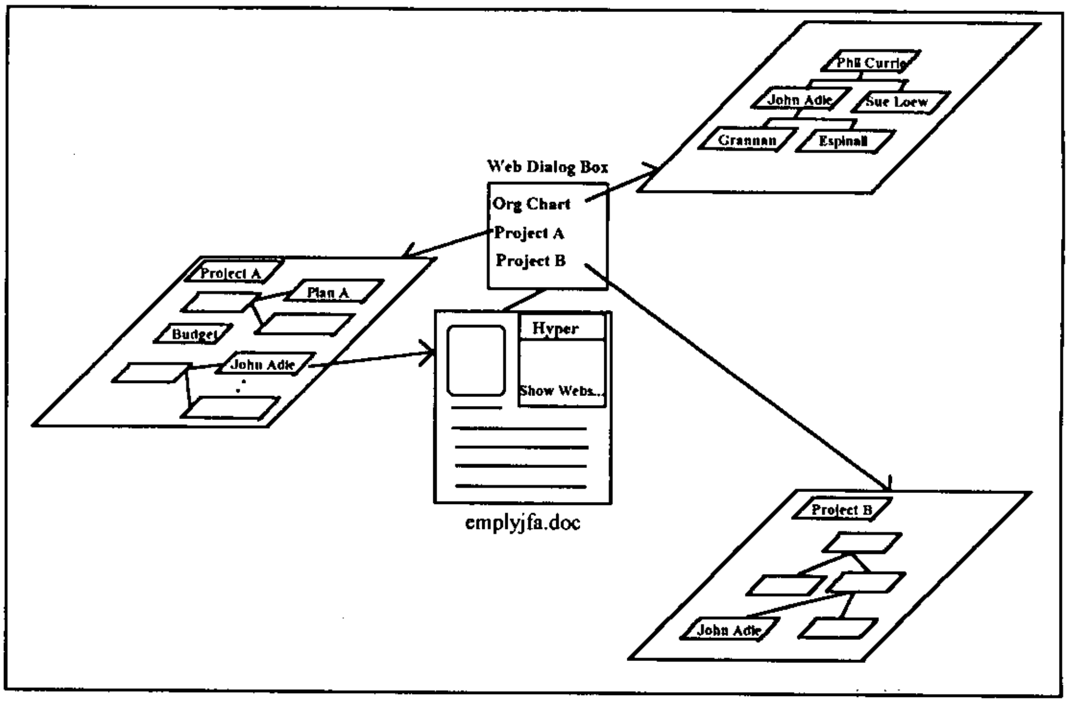


Figure 8. Webs with a Common Node

- The links represented visually on the webs exist only as graphical images in the web window. The file itself is not altered. All information pertaining to the nodes and inter-node relationships on a web is stored on the web itself.
- Webs may be assigned attributes and treated as any other node in the system. This facilitates the notion of a composite node. It also facilitates the retrieval of desired webs for subsequent file retrieval or hypertext navigation.

4.3 Technical Issues

Several technical problems emerge, some of which have been identified in other hypertext work. Some of the difficulties can be overcome at this time, and others can be worked around. We believe that there is too much power to be gained at the workstation by the inclusion of generic hypertext-type techniques to dismiss it until it can be all things to all people.

Granularity of nodes. There is a belief on the part of many hypertext researchers that defining a node at the file level is too coarse a grain and not acceptable. Our facility defines nodes for any selected range in an Excel spreadsheet, and currently defines nodes at the page level for Word files and ToolBook applications, although this can be changed to location on a page in a future version. All other files, accommodated through the Link Facility desktop accessory, have nodes defined at the file level. Interestingly, we have found that our typical end users are more likely to think at the file level anyway. The initial application users make of our features is to develop a collection of webs and browsers that might be thought of as graphical logical directories. In this context, file level nodes are quite suitable, even natural.

Ongoing changes within files. If contents and/or their location within a file change substantially, the definition of a segment of that file as a node will not necessarily be stable. In a traditional hypertext situation where there are many within-file nodes that are linked together, our prototype would not perform satisfactorily if the node sizes or locations within the file were volatile. Our users are not in the business of trying to develop hypertext documents from one file. With stable files, defining segments of the file as a node can be accommodated quite satisfactorily with our model. For example, one user found it useful to link separate files to sections of a regulations manual.

Indexing and attribute definition. This is a challenging problem, with a wide range of considered opinions on the best way to approach the question. Many developers believe that indexing techniques must be automatic, because users will not take the trouble to enter their own keywords. This is workable for textual nodes, but not for all the non-textual file types that we typically want to include. Furthermore, indexing methods are developed for large scale information retrieval needs and, although very effective, are not perfect anyway. In business situations, appropriate keywords may not even be included in the text body. In our system, a user specifies attribute values during setup, then just chooses desired values from a dialogue

box when defining node attributes. New attribute values are automatically placed on the permanent list when entered as extra values during node attribute definition. Our observations of use indicate that a user's categorization changes as a problem is understood better or as an operational situation changes. In order to accommodate recategorization, users can recall a list of all nodes with specified attribute values by query, then replace or delete that attribute for all or a subset of those nodes. It would be impossible to be able to carry out recategorization in a completely automated fashion. In fact, with automated indexing one loses the control that allows recategorization. A good compromise would be to carry out automated indexing on suitable files, compare the results to a user-specified list of keyword values, and assign those keywords to the file as default values. The default values could obviously be overwritten by our current user-controlled assignment method.

Attribute values provide two functions: one is for straightforward retrieval and the other is to form interesting associations with other nodes through various combinations of related attribute values. An issue regarding how to provide maximum effectiveness in forming associations is how many keywords need to be stored. Our system currently allows for several automatic attributes: file name, path, location, date and time of storing, last access date, application, and object type (browser, web, excel worksheet, excel macro sheet, etc.). Optional user-defined attributes include a node label (default is file name), owner, project, short description, and four keywords. For most situations, four keywords seems to be acceptable. For some situations, however, up to ten would be useful. In a second version, we will consider the performance tradeoffs of storing more keyword values.

The observation that users simply will not bother to enter attribute values on a regular basis cannot be dismissed. In fact, our experience with users to date shows that with day-to-day work they prefer to develop a collection of webs and browsers without entering attribute values, although they may assign attribute values to their webs. Eventually they encounter a situation for which it becomes apparent that assigning attribute values to a large number of nodes would be worth the trouble. However, when a new project is started and it is realized that many of the hypertext-type features will add value, the discipline of assigning attribute values is established at the outset of the project. In our vision of having such features become a part of any computing environment, determining attribute categories and keyword values would become an identified activity of systems analysis and design.

Coordinating with operating-system controlled actions. Standard operating systems keep propriety control over files, maintaining file names, renames, directory structure, file paths, and moves. As long as this is the case, some aspects of open hypertext implementation will be less than perfect. Such implementations have to keep databases (or hyperbases) that try to maintain the same data that the operating system keeps, but when moves or renames are carried out, the hyperbase does not have access to the changes. Therefore, keeping an open system hyperbase current in a volatile information base is problematic. What is required is for the operating systems to capture the attribute data, which would then remain attached to a file regardless of change in name or path. This would overcome many of the problems with current systems, including ours.

Networked workstations versus stand-alone workstations. Our facility has been installed on both networked and stand-alone workstations. It works just as easily in including nodes stored on the network as it does with nodes stored on the user's hard drive. The prototype is designed with one hyperbase; the hyperbase can be conceived of as a personal database or a shared one. It can be set up either way, with a personal hyperbase installed on each user's machine or with a central hyperbase set up to be accessed over a network. The networked version invites the standard problems of concurrency and worries of control issues such as who can change attribute values. Our current prototype takes a relaxed view toward these concerns. If the users wish to share one hyperbase, then they will decide for themselves on protocol of ownership, etc. As mentioned earlier, in our Windows prototype, Windows itself ensures integrity under concurrent use. Many of these problems are being addressed in hypertext that is being applied to collaborative work (Halasz 1988; Irish and Trigg 1989; Benest and Dukic 1993). These reported efforts, once again, are implemented as monolithic systems rather than superimposed on the existing operating system and existing applications. A future version of our prototype may concentrate on concerns associated with a shared model. Our current version is recognized by users as affording them greatly enhanced file access as a minimum and as providing a means by which to make improved use of *all* available information for decision making.

5. EXPERIMENTAL STUDY: INITIAL RESULTS

Link Facility has been developed to test several hypotheses concerning the enhancement of the decision making process through improved access methods for qualitative information, in particular through open hypertext and associated tools. The potential importance of this approach to future DSS research has been identified in the report resulting from the Information Systems and Decision Processes (ISDP) study (Stohr and Konsynski 1992). Preliminary studies have been carried out, using Link Facility in a controlled environment. An information base of 250 files that have been stored in a hyperbase, with attributes and link structures established, were used for our first set of experiments. Twenty subjects, all having a background in the subject matter to which the files pertain and all having a working knowledge of the Windows environment, were asked to use the information base to help them formulate questions, answers, and ideas for the development of a specified policy issue. The choice of menu commands used was monitored by the software and by an experimenter. The quality of the information retrieved was scored by an expert. Subject profile data was also collected. The details of these experiments and others, and complete analysis of results, are the subject of another paper; however, the subjects reported that they felt that many of the tasks required could not be done properly without the use of webs. Webs were seen to have more value than retrieval through attribute query, although both features added to the subjects' ability to make effective use of stored information.

6. CONCLUSIONS

We have described a model that ideally would see the operating system in any computing environment supporting the possibility of open hypertext on the desktop. Although hypertext is seen largely in the context of creating electronic documents, we propose the use of hypertext-like techniques at the file and subfile level to enhance the accessibility of information and the appropriateness of retrieved computer-accessible information.

Current file management tools are not always seen as effective by end users for even simple tasks such as organizing files by directory structures. In fact, many end users seem to rely largely on meaningful file names for retrieval. As soon as they introduce directories and subdirectories, they find an application will save a new file in a default directory that is a mystery to them - they do not know the location of the new file. Being able to create alternate logical directories in the form of webs, a visual representation of how *they* view their files and the relationships between them, is seen as a positive enhancement to existing file access methods.

The use of computers in business has become so pervasive, and its range of uses so much more diverse than previously envisioned, that our workstations have become the repository of a vast array of useful information in many different forms. However, our ability to make effective use of this material for qualitative approaches to decision making is not supported. For example, there may be many documents that are pertinent to a policy review, including memos, reports, scanned maps, CAD drawings, etc., but whether or not each relevant document is included in a review depends on the memory of the user. The model described in this paper, that of superimposing associative networks in the form of hypertext-like structures on a user's information base, affords the user improved accessibility and utility of their information.

7. REFERENCES

Benest, I. D., and Dukic, D. "Computer Supported Teamwork" In D. Diaper and C. Sanger (Editors), *CSCW in Practice: An Introduction and Case Studies.* New York: Springer-Verlag, 1993, pp. 127-150.

Bieber, M. P., and Kimbrough, S. O. "On Generalizing the Concept of Hypertext" *MIS Quarterly,* Volume 16, Number 1, March 1992, pp. 77-93.

Boden, M. A. *The Creative Mind: Myths and Mechanisms.* New York: Basic Books, 1991.

Booth, P. *An Introduction to Human-Computer Interaction.* London: Lawrence Erlbaum Associates, 1989.

Conklin, J., and Begeman, M. L. "gIBIS: A Hypertext Tool for Exploratory Policy Discussion." *ACM Transactions on Information Systems,* Volume 6, Number 4, 1988, pp. 303-331.

Davis, H.; Hall, W.; Heath, I.; Hill, G.; and Wilkins, R. "Towards an Integrated Information Environment with Open Hypermedia Systems." *Proceedings of the European Conference on Hypertext'92,* Milan, Italy, November 1992, pp. 181-190.

Dawson, R. *The Confident Decision Maker.* New York: William Morrow & Co., 1993.

Dreyfus, H. *What Computers Still Can't Do: A Critique of Artificial Intelligence.* Cambridge, Massachusetts: MIT Press, 1992.

Dreyfus, H., and Dreyfus, S. *Mind Over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*, Second Edition. New York: Free Press, 1988.

Haan, B. J.; Kahn, P.; Riley, V. A.; Coombs, J. H.; and Meyrowitz, N. K. "IRIS Hypermedia Services." *Communications of the ACM*, Volume 35, Number 1, 1992, pp. 36-51.

Halasz, F. G. "Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems." *Communications of the ACM*, Volume 31, Number 7, 1988, pp. 836-852.

Halasz, F., and Schwartz, M. "The Dexter Hypertext Reference Model." *Proceedings of the Hypertext Standardization Workshop*, Gaithersburg, Maryland, January 1990, pp. 95-133.

Irish, P., and Trigg, R. "Supporting Collaboration in Hypermedia: Issues and Experiences." *The Society of Text: HyperText, Hypermedia, and the Social Construction of Information.* Cambridge, Massachusetts: MIT, 1989.

Kacmar, C. J., and Leggett, J. J. "PROXHY: A Process-Oriented Extensible Hypertext Architecture." *ACM Transactions on Information Systems*, Volume 9, Number 4, 1991, pp. 399-419.

Mahapatra, R. K., and Courtney, J. F. "Research Issues in Hypertext and Hypermedia for Business Applications." *Database*, Volume 23, Number 3, 1992, pp. 10-18.

March, J. "How Decisions Happen in Organizations." *Journal of Human-Computer Interaction*, Volume 6, Number 2, 1991, pp. 95-117.

Marshall, C. C., and Rogers, R. A. "Two Years before the Mist: Experiences with Aquanet." *Proceedings of the 1992 European Conference on Hypertext*, Milan, Italy, November 1992, pp. 53-62.

Marshall, C. C.; Halasz, F. G.; Rogers, R. A.; and Janssen, W. C. Jr. "Aquanet: A Hypertext Tool to Hold Your Knowledge in Place." *Hypertext'91*, 1991, pp. 261-275.

Nielsen, J. *Hypertext and Hypermedia.* New York: Academic Press, 1990.

Parsaye, K.; Chignell, M.; Khoshafian, S.; and Wong, H. *Intelligent Databases: Object-Oriented, Deductive Hypermedia Technologies.* New York: John Wiley & Sons, 1991.

Rettig, M. "Hat Racks for Understanding" *Communications of the ACM*, Volume 10, 1992, pp. 21-24.

Shu, N. *Visual Programming.* New York: Van Nostrand, 1988.

Simon, H. A. *The New Science of Management Decision.* New York: Harper and Row, 1960.

Sternberg, R. J. "Coping with Novelty and Human Intelligence" In P. Morris (Editor), *Modelling Cognition.* Chichester, England: John Wiley & Sons, 1987, pp. 57-91.

Stohr, E., and Konsynski, B. *Information Systems and Decision Processes.* Los Alamitos, California: IEEE Computer Society Press, 1992.

Winograd, T., and Flores, F. *Understanding Computers and Cognition: A New Foundation for Design.* Reading, Massachusetts: Addison Wesley, 1987.

Yankelovich, N.; Haan, B. J.; Meyrowitz, N, K.; and Drucker, S. M. "Intermedia: The Concept and the Construction of a Seamless Information Environment." *IEEE Computer*, Volume 21, Number 1, 1988.

Young, L. F. *Decision Support and Idea Processing Systems.* Dubuque, Iowa: Wm. C. Brown, 1989.