N88-14859

S4-82 116648 128

1B 128796

A REVIEW OF HYPERTEXT IN A NASA PROJECT MANAGEMENT CONTEXT

Final Report

NASA/ASEE Summer Faculty Fellowship Program--1987

Johnson Space Center

Prepared by:	Christopher J. Bell
Academic Rank:	Assistant Professor
University & Department:	Illinois State University Applied Computer Science Dept. Normal, Illinois 61761

NASA/JSC

Directorate:	Space and Life Sciences
Division:	Man-Systems
Branch:	Crew Station
JSC Colleague:	Marianne Rudisill
Date:	August 15, 1987
Contract Number:	NGT 44-001-800

Abstract

This document discusses the principles of data storage, the comparative strengths of data bases, and the evolution of hypertext within this context. A classification schema of indexing and of hypertext document structures is provided. Issues associated with hypertext implementation are discussed and potential areas for further research are indicated.

Introduction

One of the strengths of the modern computer is its ability to manipulate large masses of data. It is anticipated that the space station will carry on board a great deal of the documentation associated with the design and construction of its components. Accessing needed information from data sets of the anticipated size of the one planned for the space station is likely to create difficulties for the user. The user may find it extremely difficult to locate the desired information or may find that the information sought is contained in a large mass of irrelevant material. Further, the ability to efficiently access most modern data bases is limited by the familiarity of the user with the peculiarities of the database organization.

Hypertext is a method for organizing text oriented data bases to facilitate ease of access, to promote rapid navigation to desired nodes, and to provide views of data paths that facilitate consideration of alternatives which might be of interest for browsing. Users find hypertext based documents facilitate browsing and simplify navigation problems that can become quite severe in other data search environments.

This paper will examine the data processing/file manipulating background against which the strengths of hypertext concepts should be viewed. Several hypertext implementations are reviewed. The author has worked extensively with one of these packages (HOUDINI by MaxThink) on a Compac Deskpro (IBM-PC XT equivalent), and some of the observations given late in the paper are the result of several implementations developed by the author using the above combination. Areas of research in human-computer interaction using hypertext that should be addressed are described.

Historic Development/Overview of Varieties of Databases

The computer has come to be viewed as a necessity in the modern world for many reasons - speed of processing, significant improvements in numeric manipulations, ability to manage large volumes of data, among many other strengths. The ability to manipulate large masses of data, to perform the same process to an endless number of records or to search a large file for the specific record needed has made the computer a mainstay of the information processing world. The first section of this paper will provide a review of record, file, and database fundamentals common to the information processing environment, and attempt to provide the reader with the background useful to understanding the virtues of hypertext.

The primary structure of information is the field. A field consists of the smallest segment of data meaningful to the user. Examples of fields might include name, home address, work address, etc. Fields alone are meaningless unless they belong to a higher data organization called a record. Each record of the same type will contain the same fields, but the contents of the fields will differ from one instance of a record to another. Associated records are grouped into files which serve as the primary organizational structure for large sets of data. An organization may maintain separate files for its employees, for its corporate resources, for its customers or students, etc.

Traditionally we have tried to organize these files into one of three structures based on the anticipated accesses which will be made to the data. If access will be to most or all of the records during any given file use, and the order of the accesses is not important or can be anticipated (i.e. alphabetical), then a simple sequential structure where one record follows immediately upon the previous will make the most efficient use of storage and reading time. Sequential storage is still the classic data organization and is found useful in many situations, but it has major deficiencies for some situations.

Many users find that they cannot predict in advance which records they are going to need or how frequently they will need to access any given record. When this is the case, files can be organized so that the location of any given record can be determined from a unique key field (such as part number) contained in each record. Thus an attempt to access a given record can be directed to the location of that record in storage, with no need to process or pass over unused records. This direct access method provides much faster access to any given record, but at the cost of wasted storage space (for locations with no corresponding record key field) and no easy way to access all records if the user so desires.

If these latter considerations are important to the user's anticipated application, one may organize the file so the records are stored immediately following one another - as in sequential - but one also maintains an index pairing the record's key field with storage location, one can have most of the storage efficiency of sequential organization with rapid access to any individual record as found with direct organizations. This indexed sequential structure provides both of these boons, but with some cost associated with maintaining and accessing the index. Most of the currently available products, even those specific to certain manufacturers (such as IBM's KSDS) can be categorized into one of these formats.

Common to all basic file organizations, however, are problems related to multiple files of similar data. As each user will have needs for data which differ somewhat from other users, each will require a file dedicated to them. Thus data is often repeated in several files, wasting space and resources. This redundancy may lead to much worse problems caused by inconsistency among files containing duplicate record fields. With many users accessing many different files, it becomes increasingly hard to maintain corporate standards of security and data integrity. Laws to control what may or may not be retained must be respected, and with diverse users and files, it becomes difficult to do so. Each attempt to initiate a new use of the computer system requires extreme investments in duplicating file services, leading to greater and greater demands on the system resources. Further if the company should wish to begin using a new computer system, file migration problems must be addressed for each application file, duplicating expenses, if even resolvable.

These disadvantages were recognized early, and many organizations have supported the concept of a database to control for many of the problems. The database is in effect a single super file containing all instances of all data, with a sophisticated management system to ensure that only legitimate users may access those parts of the data to which they have a need. Access can duplicate any structure the user is familiar with, but there is a major expense involved in maintaining and coordinating the database. The terms database management system and database administrator have been developed to reflect this cost.

As with files, three database structures have evolved. The hierarchial data structure consists of a series of root or parent nodes each pointing to several branches or descendents. By anticipating probable accesses, nodes with similar demand characterisitcs can be grouped to simplify access procedures. For example, if Part Type is the root structure, it is much easier to generate a list of all suppliers of a given part than it it is to find all parts from a given supplier. If both types of accesses are anticipated, the user can duplicate all information using the second structure (which is expensive) accept the penalty of time and inefficiency associated with the structure, or establish a second set of links between associated parts.

This later approach characterizes the network oriented database structure. Data is retained as discrete records with links connecting the various similar internal fields between records. Any record may have a large number of links coming in or going out (multiple roots/parents, multiple branches/children).

Unfortunately, both hierarchial and network structures cost the user substantial overhead, in many cases 200% - 300% of the space used to store the actual data.

The third solution to the problems associated with files that has become popular with the decline in cost of computational power is the relational database design. Records are stored as fixed relations in large tables. User requests are viewed as segments of the table, sometimes single fields, sometimes small sets of fields or records. Relational databases are generally considered superior to the othe designs because of the inherent flexibility in their design and their ability to respond to varying, unanticipated requests. Users of relational databases need to acquire some sophistication in their requests, as often requests may yield either a null set or substantially more information than is needed. The user specifies the characteristics to search the database for, and the exact matches are provided back to the user, much as a donut cutter may slice a donut from rolled out dough.

These traditional databases share some common strengths and weaknesses. Substantial overhead is needed to maintain links/structures of the database when compared to the traditional file structure. Their organization tends to favor numeric or short memo storage in contrast to lengthy text. Databases however, reduce waste due to redundant data, and can avoid (in centralized databases) or minimize (in distributed databases) inconsistencies within the data set. Data can be easily shared among various applications and users, while corporate, national, and industrial standards can be enforced. Any applicable security restrictions are easier to maintain in a database oriented environment while data integrity and accuracy can be maintained.

Springing from this file and database background, the hypertext concept has found fertile soil for the users of computer-based document systems. We shall now address the concept of hypertext.

The Concept of Hypertext/Hypermedia

Reading is fundamentaly linear. Words are grouped together to form sentences, sentences to form paragraphs, paragraphs to form documents. Each has a beginning and is read through to the end. If the reader is searching for a particular piece of information contained in the document, the document will be read (or skimmed) until the information is located, again in a linear fashion.

With large information oriented documents, such as texts, the reader may elect to use an index to locate the page containing the particular reference. The page is then read to locate the information sought. In encyclopediae and dictionaries, the user can turn directly to the location of the information sought, but the act of extraction of the information will be linear. Obviously, the finer the division of indexing, or topic selection, the less reading is needed to extract the desired information.

While the analogy may be strained, one can view a linear document as corresponding to the sequential file structure, indexed documents as corresponding to the indexed sequential structure, and the encyclopedia/dictionary as a direct access structure. In all three however, the information sought still needs to be extracted from the surrounding document matrix through the linear task of reading.

Often relevant information about a topic will be contained at more than one location in a given document. Thus one encounters the multi-page references of indexes or the "see also" entries of the enclycopedia and dictionary. These cross references take on some of the characteristics of hierarchial and network data bases.

All of these referenced text products have been reproduced in computer usable formats. The ability of the computer to rapidly process file information and to search for particular addressable locations (as in a network oriented database) may permit the text developer an extra level ofreference ability, the ability to backward reference an information item. The user may consult an index (or menu) and bring up a particular piece of information. That particular piece of information may in turn lead to other locations in the data set. At any of these further locations, the user has the option of selecting one to many further references or of "looking the other way" to see from where the current location has been referenced. By establishing a view of both the incoming and outgoing references, the user can pursue data references in unique combinations of trails. Supported by proper hardware capabilities, this concept of multi-referenced nodes of information is termed hypertext. If illustrations, graphics, sound, etc. are added, the result is a hypermedia document.

One way to understand the importance of the referencing capabilities of hypertext is to momentarily digress and create a taxonomy of access methods to obtain information inprinted documents. At the 0th level we have no indexing, and searches for information are strictly sequential. At the 1st level, we place the index pointing directly to the topic area. Further complexity of indexing is seen in the encyclopedia or dictionary level in which the index points to topics which in turn point to other topics (see also...concept). This would be termed the 2d level. If we design our information retrieval system such that indexes point to nodes which in turn point to other nodes (as in the 2d level) but the nodes carry information as to which nodes are pointing to them, so that the user may not merely backtrack but also explore alternate routings both forward and backward through the data set, we achieve the 3d level of indexing. By cross-linking our initial data set to reference footnotes, other data sets which might contain relevant information, further indexes, etc., we develop a truly comprehensive information retrieval system. Hypertext implementations are presently at this third and fourth level of functional development.

A well designed hypertext document is a fully referenced text organization. All items of potential interest are linked to explanations, related items, and other related text locations. For example, the text of poem which uses regional dialect or colloquialisms might provide a means to immediately substitute the modern expression for the regionalism (Guide demonstration disk), but might also be incorporated in turn into a data set of poems from the region, with the ability to isolate a geographic area and see a list of poetry (or whatever) associated with that area, perhaps by a graphic of a map and a freely moving cursor. In addition to poetry, expanded views of the area selected might also be optionally accessed, with historical and social commentary provided for background to the poetry.

Hypertextual organized reference materials promote fast access to the information desired. They also provide the user with a browsing capability of examining both the node sought and information about related nodes of information. This broadened picture very much strengthens the user's conceptual grasp of the data and the relations among the various nodes. Available alternatives become so rich that the problem arises of navigating through the nodes, and the concern arises that the user will become "disoriented" both as to how to continue on to their goal as well as how to return to their initial level of operation.

Hypertext Implementation Issues

As with so many other communication issues, the structure and organization of the nodes and links used in the hypertext document can reduce or heighten the disorientation problem. Proper sizing of nodes, recognition of the limits of human perceptual and cognitive limits, and other psychological and perceptual factors need to be remembered by the development team responsible for creating the document. Not all of these factors have been finally defined, particularly in regard to functional use in the hypertext environment.

In creating a hypertext document, information is collected in textual chunks sufficient to contain one idea or concept. Each of these nodes are linked to other nodes containing related material. While no absolute limit has been discussed in the literature, it has been observed by the author that more than seven links - either coming in or leaving the node - become difficult to manage for the user. Exceptions to this do occur, particularly where one is dealing with accessing a series of related cases, but for most uses this is a practical guideline.

One of the defining characteristics of a hypertext system is the presence of direct machine support for references between the nodes. This machine support implies that the user can jump from one node to another through single keystrokes or cursor movement/keystroke activities.

Hypertext concepts have been extended to several areas. The first of these is the on-line library or literary system. The documents in the library would be linked by machine supported hypertext nodes. Provision is made for users to add comments or criticisms and to respond to others' comments. Document creation and collaborative efforts can be supported through the underlying software systems. A current implimentation of this is available from McDonald-Douglas called NLS/Augment and is in active use with the Air Force.

A second category of hypertext implimentations are problem exploration tools. Because of the ability of hypertext to handle ideas and to quickly link those ideas, parallel concept generation is easily supported. Multiple ideas about a topic - or set of topics - can be created, with the author able to remain unconcerned about the relationships among the ideas during the initial creative stage. The linearity of thought required for traditional text generation can be bypassed during the initial creative rush. An example of these problem exploration tools is Maxthink from MaxThink Corporation.

The third category of hypertext systems are the browsing systems. These read-only systems are useful for teaching, reference and information systems. ZOG from Knowledge Systems, Inc. has been implimented as an intelligence review system for the Navy. The Interactive Encyclopedia (TIE) under development at the University of Maryland is a second example.

Most of the developed hypertext systems reviewed by the author fall into one of the three categories described above. Common to all is the presence of machine supported intertext references that permit the user to move forward and backward through the text. A mechanism for automatically marking the nodes or "trail" used through the document is also a common feature. Many permit backward tracking to exit from a particular node trail, and most also permit a quick exit to some form of main menu or initial state to regain the "large picture". All but the purest of browsing systems allow the user to add annotations or links to the document to customize a general system to their individual needs.

Screen clutter becomes a problem as the number of nodes accessed grows. The user will often feel disoriented due to the many degrees of freedom available for movement at any point in a moderately complex hypertext document. To solve this different approaches have been implemented. Overlapping windows of text permit the user to see the most current text while leaving evidence of other windows present on the screen. Unfortunately, as the size of the current window is increased to accomodate increased text or to improve legibility, the evidence or presence of other windows disappears, requiring the user to backtrack until a cue to sigificantly earlier windows becomes visible.

Several implementations use an icon approach to maintain user awareness of alternate pathways. It can become a significant task for the user as well as for the developer to establish sufficiently unique icon designs to distinguish among many alternatives available, and to recognize the appropriate icon to select next. Both appropriate symbols and efficient abbreviations still require user training for maximum effective performance. Other approaches presently being studied involve rich graphic fields using rooms/doors/rooms analogies and flight analogies. Whether these approaches will prove effective given their relatively high cost in system overhead is still being examined.

Areas for research and related topics

A key to increasing the impact of hypertext on computer-based document systems in the future is the improvement of the user interface to take advantage of the innate strengths while minimizing the effects of the innate weaknesses present in the user. How should the presence or absence of further links or help alternatives be indicated? Is the use of special printing characters such as reverse video or special characters an appropriate technique or should icons be attached in some fashion to linked text areas? Should these be always visible or should they become visible with a keystroke (togglable)? How natural is the use of a hypertext designed document? Should naive users be expected to quickly find their way through it or will some training or orientation be required?

We know that users possess varying abilities of spatial orientation and that these abilities can simplify the task of managing the data relationships present in a hypertext document. Users are able to retain several stages of nodal levels in their memory and are able to handle visual fields containing approximately thirty items before performance degrades significantly. Designers of hypertext documents may be able to work with these limits in the development of their work.

Areas of immediate importance to NASA for evaluating the utility of hypertext for space flight activities include a comparison between hypertext and traditional computer-based document storage systems. Specifically, training time to a given level of functionality, error rates during test trails, retraining needs after a significant non-rehersal time period, and graphic versus textual presentation approaches will need to be examined.

Summary

Hypertext represents a fascinating fieldfor exploration of the relationships between users, data, and computers. Work is proceding in using concepts from hypertext in fields as diverse as education and expert systems. Several commercial packages are being made available in the immediate future. It is my hope that the reader will continue to explore this subject area and pass along to the author any observations or products which should be found.

BIBLIOGRAPHY

Robert Akscyn and Donald McCracken, "ZOG and the USS CARL VINSON: Lessons in System Development",Carnegie-Mellon Technical Report No. CMU-CS-84-127, March 1984.

Stuart Card and Austin Henderson, "A Multiple, Virtual-Workspace Interface to Support User Task Switching", Intelligent Systems Laboratory, Xerox Palo Alto Research Center, Palo Alto, California 94304.

Jeff Conklin, "A Survey of Hypertext", MCC Technical Report No. STP-356-86, Rev. 1, February 1987.

Donna Erb, "Trip Report - May 18 - 22, 1987 JSC Missions Operations Training Task", MITRE Document

George Furnas, "Generalized Fisheye Views", Bell Communications Research, Morristown, New Jersey.

Guide User's Manual, Owl International, Inc., 144218 NE 21st Street, Bellvue, Washington,98007.

W.P. Jones and S. T. Dumais, "The Spatial Metaphor for User Interfaces: Experimental Tests of Reference by Location Versus Name",ACMTOOIS, Vol 4, No. 1, pp. 42-61, January 1986.

Neil Larson, MaxThink Journal, MaxThink, Piedmont, California, Vol. 2:3, January 1986.

Neil Larson, Houdini 2.1 Manual, MaxThink, Piedmont, California 94610

Jeffrey Young, "Hypermedia", Macworld, March 1987