

UNIX and X-Windows Implementations for the Hyperties Hypertext System

Proposal to UMIACS Lab Committee

from

**Human-Computer Interaction Laboratory
Department of Computer Science**

Ben Shneiderman

December 12, 1988

Abstract: We have been developing the PC Hyperties electronic encyclopedia (a hypertext system) for six years and the SUN NeWS version for three years. While there have already been practical applications in museums, education, and information retrieval we continue to refine our software and seek workstation support for reimplementing in the UNIX and X-Windows environments. We seek a powerful machine that supports X-Windows, preferably with color.

After the initial effort of producing a text-only browser, future possibilities include dealing with graphics and producing an author tool. Then the support for advanced features to serve multiple users becomes an exciting possibility. Additional directions for future research include: an advanced browser with powerful search facilities, multiple window strategies, automatic importation of existing files, an automatic exportation facility, a graphic browser, etc.

Introduction

The hypertext approach that we have been developing for six years supports rapid and convenient traversal of large non-linear databases of text and graphics by merely pointing at phrases or graphic image components. The embedded menu style of interaction has proven to be an effective means for people to locate required information in large and complex text and graphic databases (many papers are available).

In early 1988, Larry Davis had promised to support our efforts providing a SUN 3/60 by May or June, but unfortunately those plans were delayed. I decided to submit a proposal as part of the UMIACS Lab Committee's request, but I hope that our earlier understanding is still in place and that we will receive a machine when funds become available. We would like a powerful workstation such as a SUN 4 with X-Windows, preferably with color.

Hyperties (Hypertext based on The Interactive Encyclopedia System) enables users to easily traverse a database of articles and pictures by merely pointing at highlighted words in context. This embedded menus approach and the simple user interface enables users to tap the substantial power of hypertext systems for browsing and information search tasks. The commercial version of Hyperties runs on an IBM PC while the research version runs on the SUN 3 workstation using NeWS. Our single SUN 3 is shared by 6 developers and students working on other efforts as well. This machine is painfully slow in running the NeWS environment and we have put in a request for the Computer Science Department to provide more memory to reduce the page thrashing phenomenon that causes delays of 30 seconds to 4 minutes to open a window!

☞ In this request to UMIACS we hope to broaden the audience for our hypertext system by developing facilities in standard UNIX environments and in X-Windows. We currently have support from NCR for the NeWS effort and have some hopes of getting support from NCR or Hewlett-Packard for the X-Windows environment. There is a very large audience for a standard UNIX implementation and we believe that with only modest effort we can build a text-only version of the browser.

Hyperties can be used to scan organizational policy manuals, as a tool for diagnostic problem solving, as an environment for creating checklists for complex procedures, in online help, to browse computer program text and documentation, as a public information resource, or to explore cross referenced materials such as legal briefs, or technical documents. The Space Telescope demonstration (seven minute videotape is available) supports two large independent windows in which users can traverse the 23 articles and graphics that have been entered from NASA sources.

Hyperties allows users to explore information resources in an easy and appealing manner. They merely point at topics or picture components that interest them and a brief definition appears at the bottom of the screen. The users may continue reading or ask for details about the selected topic. An article about a topic may be one or more screens long and contain several pictures. As users traverse articles, Hyperties keeps the path and allows easy reversal, building confidence and a sense of control. Users can also select articles and pictures from an index.

Initial plan of work

1) UNIX Support

Our initial task will be to develop a text-only UNIX browser that would run on a wide variety of terminals with keyboard support for cursor motion. When this initial task is complete we could proceed (depending on funding) to supporting graphics and then develop authoring facilities. We expect that the UNIX author will be a programmer-oriented facility with UNIX-like commands that utilizes the UNIX file environment and other tools.

2) X-Windows Support

X-Windows has widespread support in the industry and we have often been asked for an X-Windows version of Hyperties. Having hardware support would enable us to pursue this direction and simultaneously learn about the interface toolkits and widgets in X-Windows. Then we would be in a better position to request support from outside sources.

Future directions for our research

3) Multi-user Support

One limitation of the current design is that it supports only single user environments. This is acceptable in read-only situations, but when annotation or collaborative authoring is needed, a new design is necessary. Multiple user annotation of articles is the easier task and this can be developed first. Merging annotations from several sources is also needed.

Then we can explore the more complex issue of multiple user support for authoring. We could begin with simple cases:

- merging existing hypertexts with no cross references
- merging existing hypertexts with compatible cross references
- merging existing hypertexts with incompatible cross references that require resolution.

We should also be able to deal with the inverse operation of splitting a hypertext into two pieces. In our environment, this is relatively easy to implement.

The final step is to permit concurrent editing of a hypertext by multiple users. This is more complex and we will have to address the granularity of locking carefully.

4) Advanced Browser

We now seek to develop an advanced browser for knowledgeable personnel who are regular users of information systems. In addition to the currently supported embedded menus, alphabetical index of article titles, full path history, and string search capability over the entire database, we propose to add backward and forward citation pointers for each article, search within a neighborhood of a node, strategies for multiple levels of indexes, opportunity to annotate articles, bookmarks, and tours.

Our thinking has been guided by the information seeking framework (see Marchionini & Shneiderman, 1988) which identifies the determinants of successful search through information resources. One major component is the power available to locate information relevant to solving specific problems while another component is the convenience in exploratory browsing. Although paper-based information resources (user guides, procedures manuals, training materials, etc.) are quite effective in many situations, there is now a grand opportunity to build powerful multi-media computer-based information resources utilizing the hypertext concept.

Our studies have shown that users can learn novel search mechanisms such as the embedded menus and that as expertise grows, the desire for more powerful mechanisms emerges. Experience with the Symbolics Document Examiner also shows the utility of more powerful search mechanisms. Our initial list of search mechanisms is derived from experience with a variety of systems, but as the research progresses we expect to find novel and valuable additional methods.

5) Invoking Procedures by Embedded Menu Selection

Selecting an embedded menu item currently produces a text, graphic or videodisk image,

but the idea could be generalized to produce a procedure invocation. This would permit movement to a spreadsheet, graphics package, simulation, etc. Another important possibility is that the hypertext system becomes the user interface front end for almost any application. Each embedded menu selection could produce a command language operation, fill in part of a form, or make selections from traditional menu systems. A major application would be to provide easier to learn interfaces, which are less error prone, and potentially even faster to use if a rapid pointing device such as a touchscreen is used.

A further extension would be to develop a full control language to describe complex sets of operations, in the style of HyperTalk in the HyperCard application on the Macintosh. Expansion in the direction of a User Interface Management System is a possibility.

6) Multiple Window Strategies

The single window strategy in Hyperties on the IBM PC and many other systems is convenient and effective for many tasks. However as task complexity grows and users must consult multiple sources of information, a multiple window approach becomes more attractive. Existing multiple window strategies are usually user-managed and encourage the cluttered desk model of overlapped windows. This is sometimes helpful, but in many cases the overhead in managing the screen space slows work and interferes with concentration on the task. We have developed a series of low cognitive load strategies for dealing with multiple windows (see paper with Lifshitz) and would like to continue this line of research. The prototypes we created were not woven into the main code of Hyperties. We propose to apply what we have learned from our usability studies and support a variety of these multiple window strategies. The strategies we like most include the independent non-overlapping windows with user choice for placement, task-related window sets, the least recently used window strategy, and the automatic rotation idea.

7) Automatic Importation of Existing Files

A persistent issue with hypertext systems is the importation of existing files. While many hypertext systems support the loading of individual ASCII files, we feel that there is a possibility of loading entire databases and providing partially automatic link marking. Working with Richard Furuta, we have accomplished in some prototypes but more experience and general tools seem necessary. His initial approach is to write a grammar to describe the incoming data and the target structure plus a mapping between the two. Then a parser or compiler-compiler package (such as LEX and YACC) could be used to carry out the process. We plan to explore this direction during the coming year, but feel that there will be many exciting directions for this research idea and practical application.

8) Automatic Exportation to Linear Print Formats

Once a large hypertext database has been created there may still be a need to have a linearized paper version. We have begun to explore possibilities in connection with a commercial database on the IBM PC, and we would like to apply these ideas in the richer and more powerful UNIX world. We believe that with modest effort we could support several strategies for

automatic exportation into TEX or troff files that could be used to produce laser printer output.

One approach is to require the author to specify a linearization of the database by building a file with each article title appearing at least once. Then there are several options for the formatting of individual articles and the construction of paper indexes. We would provide software support for this user managed approach and permit flexibility in formatting.

A second approach is to explore ways in which a computer-based algorithm could automatically provide some meaningful organization. Our initial hypothesis is that we could begin with the root article (introductory article) and then use information about incoming and outgoing pointers to automatically provide a useful linearization. We would like to explore possibilities using the 30-40 databases that are available to us.

9) Graphic Browser

Hyperties currently offers an alphabetic list of article titles as a form of index. In several projects we have added a Table of Contents to provide a meaningfully organized overview in the form of hierarchically clustered article titles. Even with the addition of string search and citation pointer lists, many observers have indicated a desire for a graphic browser that shows the node and link structure of articles and references. This is done in NoteCards, the Symbolics Document Examiner and other hypertext projects. While this does have appeal with small databases, there is a danger of overcrowding and confusion beyond 50 nodes. However, it does seem possible to explore strategies to gain the apparent benefits of a graphic overview while avoiding the disadvantages of a cluttered display. Clusters or modules of nodes would afford some advantage as might zooming out or zooming in. We have begun to explore automatic algorithms for structuring such networks and semi-automatic strategies that require human input.