A Short Course
on
Hypertext

John Leggett
John L. Schnase
Charles J. Kacmar

Hypermedia Research Lab

JANUARY 1989

TAMU-HRL 89-004

Table of Contents

Table of Contents

A Short Course on Hypertext

John Leggett
John L. Schnase
Chuck Kacmar

Introduction

This report contains a set of overheads for a short course on hypertext. The material has formed the basis of several introductory workshops on hypertext/hypermedia systems. The short course is under constant revision as research is published in hypertext and related areas. We welcome any suggestions that contribute to the accuracy and completeness of the information contained in this report.

A short history of hypertext

People

- Vannevar Bush (1890-1974)
- Douglas Engelbart (1925-)
- Theodor Holm Nelson (~1938-)
- Andries van Dam
  - Steve Feiner
  - Norm Meyrowitz
- Others
  - Alan Kay
  - Allen Newell
  - Rob Akscyn & Don McCracken
  - Randy Trigg
  - Frank Halasz (Trigg, Moran)
  - Mayer Schwartz & Norm Delisle
  - Bruce Schatz & Michael Caplinger

Vannevar Bush (1890-1974)

1916 - 2 Ph.D.'s from Harvard and MIT (Engineering); Married

1932-1938 - VP of MIT and Dean of the School of Engineering

1938-1955 - President of Carnegie Institute

1941 - Influenced establishment of the Office of Scientific Research and Development (OSRD) and became the Director; (WW II and the Manhattan Project)

1951 - Influenced the establishment of the National Science Foundation (NSF)

1955-1962 - Chairman of the Board of Merck and Co.

In 1945 Bush wrote an article for the July issue of The Atlantic Monthly entitled "As We May Think." This article has become the seminal paper in the area of Hypertext. To handle the information explosion occurring in 1945 and to give scientists a goal to achieve after their wartime efforts, Bush proposed the development of a special machine he called the "MEMEX."

- Bush, Vannevar. 1945. As We May Think. *Atlantic Monthly*, Vol. 176, (July), 101-108.

Vannevar Bush (cont.)

"Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing ... "

"The human mind operates by association ... in accordance with some intricate web of trails ... "

MEMEX (MEMory EXtender)

- Individual use ... a mechanized private file and library ...
- Desk with slanting translucent screens, keyboard, buttons, levers (microfilm)
- Several projection positions
- Marginal notes and comments easily added
- The essential feature is associative indexing ... (selection by association)
- The process of tying two things together is the important thing ...
- Trailbuilding and trailblazers

"The inheritance from the master becomes, not only his additions to the world’s record, but for his disciples the entire scaffolding by which they were erected."

"Materials come ready made with a mesh of associative trails ready to be dropped into the MEMEX and there amplified."

Douglas Engelbart (1925-   )

1945 - Naval radar technician, read Bush's article in a Red Cross library in the Philippines

1951 - Entered graduate school at UC-Berkeley; started designing "Augmented systems"

1957 - Ph. D. Berkeley in electrical engineering; started at Stanford Research Institute (SRI)

1962 - Published "Augmenting Human Intellect: A Conceptual Framework"; H-LAM/T

1963 - ARPA began funding Augmentation Research Center (ARC) at SRI; NLS development; Bootstrapping

1965 - Developed the mouse and chord keyset

1968 - NLS (1st Hypertext system) demonstrated at FJCC (mouse, chord keyset, windows, associative indexing, real-time distributed collaborative work environment, journal, etc.)

1973 - Augmented Knowledge Workshop (toolsets for the knowledge worker)

1975 - ARPA dropped funding for ARC (consider influence of personal computers)

1977 - SRI sold ARC to Tymshare and renamed NLS to Augment; Engelbart left also

1984 - McDonnell Douglas bought Tymshare and started the Augmentation Systems Division

1989 - Stanford has set up an Augmentation Project with Engelbart as Director

Doug recognized, as did Bush, that the amount of information was growing at an overwhelming rate. He also realized that the complexity of the information was growing even faster ...

- Engelbart, D.C. 1963. A Conceptual Framework for the Augmentation of Man's Intellect. Howerton and Weeks (Eds). Spartan Books. In *Vistas in Information Handling,* Washington, D.C., pp. 1-29.
- Engelbart, D.C. 1984. Collaboration Support Provisions in AUGMENT. OAC '84 Digest, *Proceedings of the 1984 AFIPS Office Automation Conference* (Los Angeles, CA., February 20-22), pp. 51-58.
- Engelbart, D.C. 1984. Authorship Provisions in AUGMENT. COMPCON '84 Digest, *Proceedings of the 1984 COMPCON Conference* (San Francisco, CA., February 27-March 1), pp. 465-472.

Theodor Holm Nelson (~1938-)

1959 - B.A. in Philosophy, Swarthmore

1960 - Designed and partially implemented a writing system as a term project in 1st programming class (graduate school at Harvard)

1961 - M.A. in Sociology, Harvard

1962 - Communication Research Institute (negative influence of existing CAI lead to links and chunk style Hypertext)

1963 - Lilly's Dolphin Research Lab

1964 - Sociology instructor at Vassar

1965 - Published 1st paper, "A File Structure for the Complex, the Changing, and the Indeterminate"; defined Hypertext and Hypermedia

1967 - Researcher at Harcourt, Brace & World Publishers (named Xanadu)

1968 - Consultant to Bell Labs on text processing systems; consulted with A. van Dam at Brown

1969 - Software design and development of Xanadu; published "A Hypertext Editing Sytem for the /360"

1973 - Graduate school and instructor in graphics at University of Illinois - Chicago

1974 - Published Computer Lib / Dream Machines

1977 - Software design and development of Xanadu; published "Home Computer Revolution"

1980 - Editor of *Creative Computing*

1981 - Senior Technical Writer at Datapoint; published "Literary Machines"

1984 - Development of Xanadu and XOC, Inc.; rewriting/updating "Computer Lib / Dream Machines" and "Literary Machines"

1988 - Xanadu sold to AutoDesk, Inc.; Ted is Distinguished Fellow at AutoDesk

- Nelson, Theodor. 1965. A File Structure for the Complex, the Changing, and the Indeterminate. *Proceedings of the 20th National ACM Conference*. 84-100.
- Carmody, S. W. Gross, T. Nelson, D. Rice, & A. Van Dam. 1969. A Hypertext Editing System for the /360. *Pertinent Concepts in Computer Graphics, Proceedings of the Second University of Illinois Conference on Computer Graphics*, (M. Faiman & J. Nievergelt, Eds.), University of Illinois Press, 291-330.
- Nelson, Theodor. 1987. *Literary Machines*. MicroSoft Press.

Andries van Dam

1959 - Went to school at Swarthmore; acquaintance of Ted Nelson

1967 - met Ted at SJCC; decided to work together on Hypertext

1968 - Designed and implemented the Hypertext Editing System (HES),
(360 Assembly Language); IBM support;
later sold by IBM to Apollo Mission Team at NASA/JSC for documentation;
met Engelbart at FJCC and saw demo of NLS

1969 - Designed and implemented the File Retrieval and Editing System (FRESS);
published "A Hypertext Editing System for the /360"

1979 - Electronic Document Project

1982 - Interactive Graphical Document Project (IGD) (Steve Feiner)

1983 - IRIS - Intermedia (Norm Meyrowitz)

- Carmody, S. W. Gross, T. Nelson, D. Rice, & A. Van Dam. 1969. A Hypertext Editing System for the /360. *Pertinent Concepts in Computer Graphics*, *Proceedings of the Second University of Illinois Conference on Computer Graphics*, (M. Faiman & J. Nievergelt, Eds.), University of Illinois Press, 291-330.
- Feiner, S., S. Nagy, & A. Van Dam. 1982. An Experimental System for Creating and Presenting Interactive Graphical Documents. *ACM Transactions on Graphics*, Vol. 1, No. 1, (January), 59-77.
- Yankelovich, Nicole, Bernard Haan, Norman Meyrowitz, & Steven Drucker. 1988. Intermedia: The Concept and the Construction of a Seamless Information Environment. *Computer*, (January), 81-96.
- Garrett, L. Nancy, Karen Smith, & Norman Meyrowitz. 1986. Intermedia: Issues, Strategies, and Tactics in the Design of an Hypermedia Document System. *Proceedings of the Conference on Computer Supported Cooperative Work*, Austin, Texas, (December), 163-174.
- Smith, Karen, & Stanley Zdonik. 1987. Intermedia: A Case Study of the Differences Between Relational and Object-Oriented Database Systems. *Proceedings of the Conference on Object-Oriented Programming Systems and Languages*, (October), 452-465.
- van Dam, Andries. 1988. Hypertext ’87: Keynote Address. *Communications of the ACM*, Vol. 31, No. 7, (July), 887-895.

Alan Kay

1977 - Dynabook concept

- Kay, A. & A. Goldberg. 1977. Personal Dynamic Media. *IEEE Computer*, (March), 31-41.

Allen Newell

1972 - ZOG 1

1975 - ZOG 2

1980 - ZOG 3

Rob Akscyn & Don McCracken

1981 - KMS

- Akscyn, Robert, D. McCracken, & E. Yoder. 1988. KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations. *Communications of the ACM*, Vol. 31, No. 7, (July), 820-835.

Randy Trigg

1983 - Textnet

- Trigg, Randall. 1983. *A Network-Based Approach to Text Handling for the Online Scientific Community*. PhD Dissertation. University of Maryland.
- Trigg, R. H., and Weiser, M. 1986. TEXTNET: A network-based approach to text handling. *ACM Trans. Off. Inf. Syst.*, 4, 1, (January), 1-23.

Frank Halasz (Trigg, Moran)

1983 - NoteCards

- Halasz, Frank, T. Moran, & R. Trigg. 1987. NoteCards in a Nutshell. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI*), April.
- Halasz, Frank. 1988. Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems. *Communications of the ACM*, Vol. 31, No. 7, (July), 836-852.

Mayer Schwartz & Norm Delisle

1984 - HAM & Neptune

- Delisle, Norman, & M. Schwartz. 1986. Neptune: a Hypertext System for CAD Applications. *Proceedings of the ACM International Conference on the Management of Data (SIGMOD*). 132-143.
- Delisle, Norman, & M. Schwartz. 1986. Contexts - A Partitioning Concept for Hypertext. *ACM Transactions on Office Information Systems (TOOIS*), Vol. 5, No. 2, (April), 168-186.
- Campbell, Brad, & Joseph Goodman. 1988. HAM: A General Purpose Hypertext Abstract Machine. *Communications of the ACM*, Vol. 31, No. 7, (July), 856-861.

Bruce Schatz & Michael Caplinger

1985 - Telesophy

- Schatz, B. R. 1985. Telesophy. Report #1 of the Telesophy Project, Bell TR #TM-ARH-002487, Bell Communications Research, Morristown, N.J.
- Caplinger, Michael. 1987. An Information System Based on Distributed Objects. *Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA*), (October), 126-137.
- Schatz, B. R., and Caplinger, M. A. 1989. Searching in a hyperlibrary. *Proceedings of the Fifth International Conference on Data Engineering*, (Los Angeles, Calif., February), pp. 188-197.

Chronology

1945	Bush	- MEMEX
1962	Engelbart	- Augmenting Human Intellect
1964	Engelbart	- ARPA funds ARC at SRI
1965	Nelson	- Defined *Hypertext* & *Hypermedia*
1968	Engelbart	- NLS demo at FJCC (1st Hypertext System)
	Nelson, van Dam	- HES
1969	van Dam	- FRESS
1972	Newell	- ZOG 1
1974	Nelson	- Computer Lib / Dream Machines
1975	Newell	- ZOG 2
1977	Kay	- Dynabook concept
1979	van Dam	- Electronic Document Project
1980	Newell	- ZOG 3
1981	Nelson	- Literary Machines
	Akscyn & McCracken	- KMS
1982	Feiner	- IGD Project
1983	Trigg	- Textnet
	Halasz	- NoteCards
	Meyrowitz	- IRIS & Intermedia
1984	Schwartz & Delisle	- HAM & Neptune
1985	Schatz & Caplinger	- Telesophy
1987		- Hypertext ’87
		- HyperCard
		- Guide
		- HyperTIES
1988	Nelson	- Xanadu sold to AutoDesk
1989	Engelbart	- Augmentation Project at Stanford
		- Hypertext ’89

Basic concepts and definitions

A new vocabulary is emerging along with the increased interest in hypertext. As is often the case, there is presently no consensus on the precise meaning of many terms used to describe hypertext and hypertext system functions. This section contains the set of working definitions used in the Texas A&M University Hypertext Research Lab [also see Schnase *et al.* 1988 and Kacmar *et al.* 1988]. These definitions provide a framework for conceptualizing hypertext architectures. A layered hypertext system architecture is also proposed as a model for organizing discussions of system functionality and design. In addition, an application-oriented taxonomy of hypertext systems is provided which helps characterize existing systems.

The goal in developing these definitions has been clarity, completeness, and conceptual simplicity. Whenever possible, the established meaning of existing terminology is preserved. New terms are introduced only when doing so helps to achieve the goal. The present list is by no means complete, and it is assumed that increasing knowledge about hypertext will lead to greater clarity and rigor in its terminolgy.

Historical Definitions

Hyper-. "... as used here connotes extension and generality. The criterion for this prefix is the inability of ... objects to be comprised sensibly into linear media ... ." [Nelson 1965].

Hypertext. "... a body of written or pictorial material interconnected in such a complex way that it could not conveniently be presented or represented on paper." [Nelson 1965]. " By hypertext, I simply mean non-sequential writing." [Nelson 1987].

Hypermedia. "Films, sound recordings, and video recordings are also linear strings. But these too can ... be arranged as non-linear systems ... for editing purposes, or for display with different emphasis. The hyperfilm [... for example] is only one of the possible hypermedia that require our attention." [Nelson 1965]. "Presumably hypertexts may contain graphics of various kinds ... ." [Nelson 1967:Note 6].

The term hypermedia, as it is commonly used, implies that information may be present in hypertext systems in forms other than text. It is clear that Nelson intended "hypertext" sufficient to describe the range of media these systems would eventually manage.

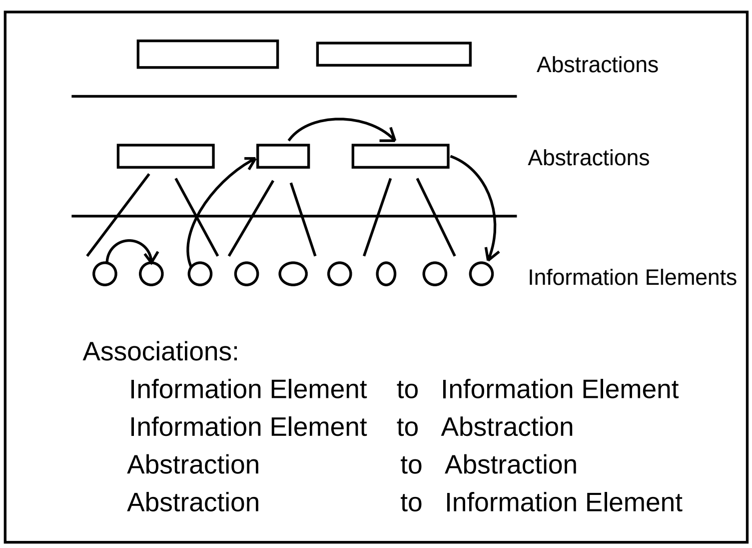
Components of Hypertext

Information element. Words, thoughts, images, sounds, knowledge, facts, data, space, time, processes, etc. Information elements may be typed and named.

Abstraction. A conceptual representation for structuring associations among information elements or abstractions. Abstractions may be typed and named.

Anchor. Anchors designate information elements or abstractions that may be the source or destination of a link. Anchors may be typed and named.

Link. A connector among anchors. Links may be typed and named. In the simplest case, links may connect anchors that are attached to a single information element or abstraction. Four basic associations are thus possible: information element to information element, information element to abstraction, abstraction to abstraction, and abstraction to information element.



Four basic associations among information elements and abstractions in hypertext.

Hypertext System Definitions

Hypertext model. A set of abstractions which provide a conceptual framework for creating, storing, and retrieving information in a hypertext.

Hypertext system. A functionally related set of computer hardware and software components that implement a hypertext model.

Author. The component of a hypertext system that creates hypertext.

Browser. The component of a hypertext system that retrieves information contained in hypertext.

Node. A generic term for an abstraction in a hypertext model.

"Include" link. A link which when followed causes the node containing the source anchor to be modified. Implies node composition.

"Automatic include" link. An "include" link which the hypertext system follows automatically when the node containing the source anchor is accessed. Implies node composition.

Composite node. A node containing one or more "include" or "automatic include" links.

Other Definitions

Content search. The content of nodes in the hypertext are searched for a match to a given query, for example, "all nodes containing 'INTERTWINGLED' ... ", or "two columns of text with a picture in the upper right-hand corner ... ."

Structure search. Examines subnetworks of nodes within a hypertext searching for a given pattern, for example, "all nodes of Type X that are connected to nodes of Type Y ... ."

Temporal search. Searches nodes in the hypertext based on a temporal attribute, for example, "all nodes that are animations of greater than five minutes ..." or "the current version on <date> at <time> ... ."

Spatial search. Searches nodes in hyperspace based on their *n*-dimensional location.

Versioning. Maintaining a history of changes to the components of a hypertext.

Trail. A path of associations through a hypertext.

History. The trail created while browsing hypertext.

Overview. A graphical representation of a hypertext.

Cache. Retaining nodes in main memory in order to reduce access time.

Extensible. The ability to extend the functionality of a hypertext system through the use of a programming language.

Tailorable. The ability to customize a hypertext system by changing attributes of the system.

View. The components of a hypertext that are accessible. A web.

Filter. The process of generating a view.

Collateral display. The ability to display nodes side-by-side.

Stretchtext. A form of hypertext consisting of "ordinary continuous text that can be ’stretched’, or made ... more detailed" [Nelson 1967:Note 6]. Selected stretchtext may be "unstretched," returning the user to a previous, less detailed context. "Thus it is unlike conventional hypertexts having discrete chunks and breaks ... " [Nelson 1967:Note 6] .

"Stretch" link. A form of "include" link that is used in stretchtext.

Models of existing hypertext systems

The following figures demonstrate how associations among information elements and abstractions (see the "Basic concepts and definitions" section of this paper) may be used to express the data models implemented by the major hypertext systems presently available. For a more complete description of each of these systems, the reader should refer to *A Comparison of Hypertext Systems* [Schnase *et al.* 1988].

The systems presented are:

- KMS
- NoteCards
- HyperTIES
- HyperCard
- Guide
- Intermedia
- Xanadu
- Augment
- HAM
- Concordia / Document Examiner
- IGD

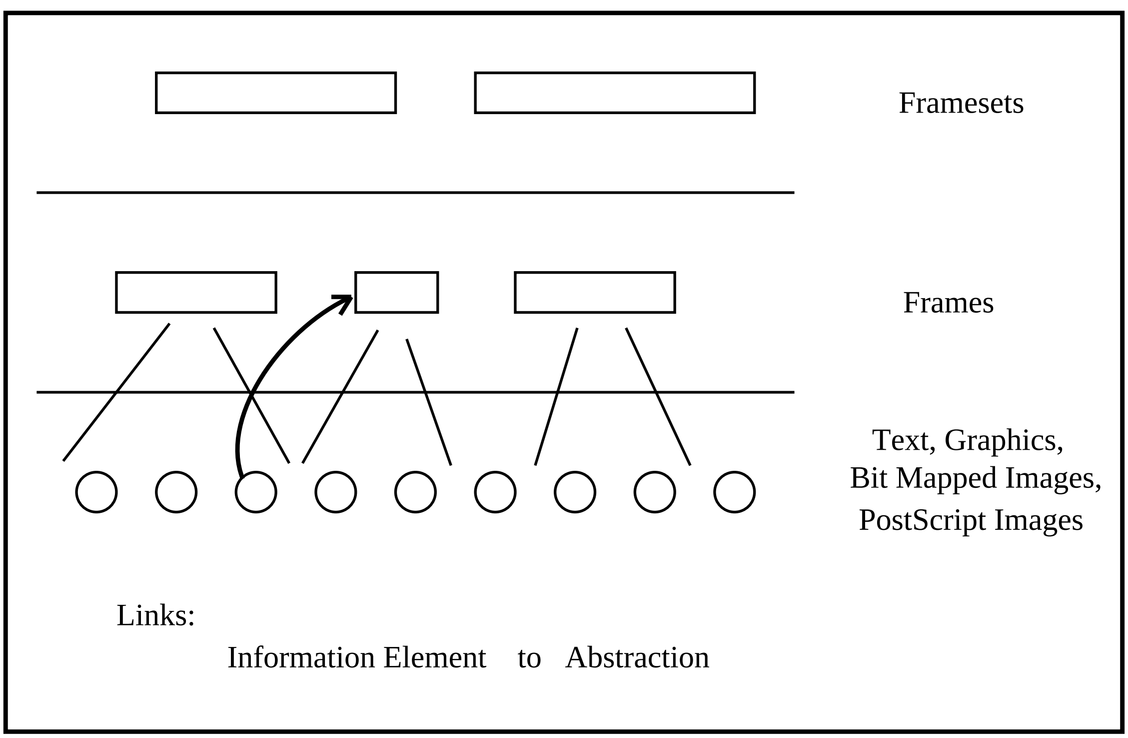
KMS

Robert Akscyn, Donald McCracken
Knowledge Systems, Inc.
1981

Data model and underlying metaphor. A KMS (Knowledge Management System) database consists of a set of interlinked, screen-sized workspaces called frames. Groups of frames are organized in framesets.

Although KMS databases may have any structure their creators desire, most have a strong hierarchical orientation. There is usually a multilevel hierarchy that acts as a skeleton for the entire database, and top levels of the hierarchy serve primarily as indexes to the entire database.

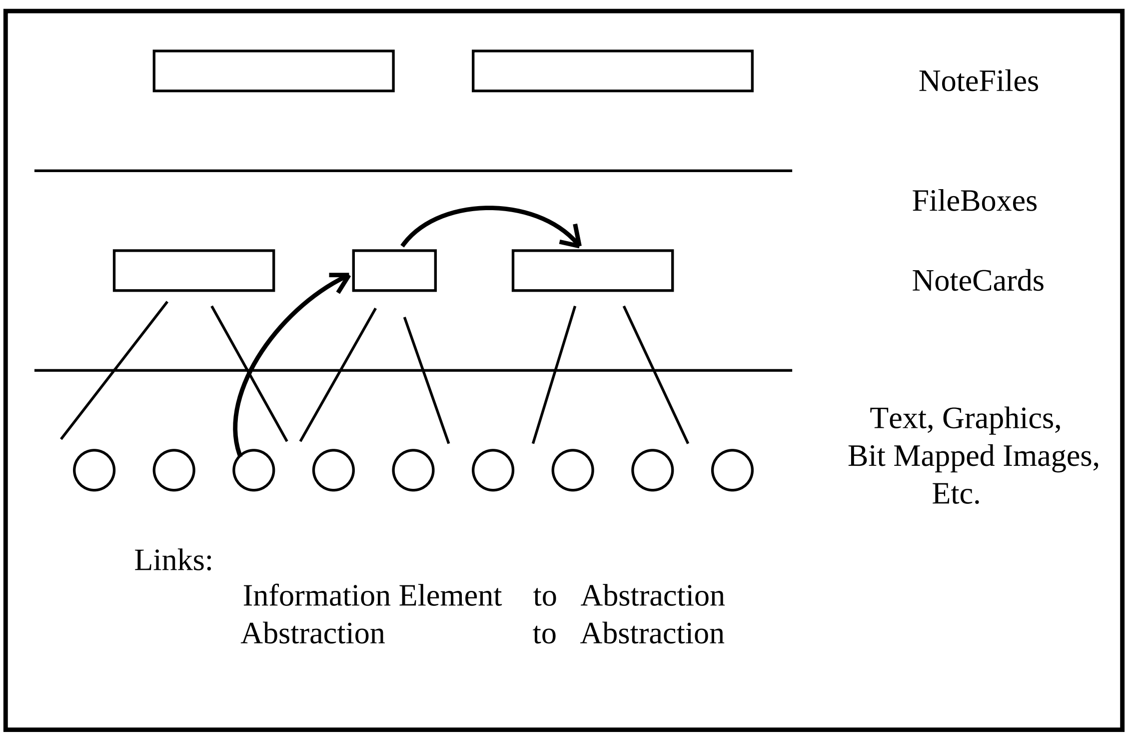
The central KMS metaphor is a universe of connected spaces through which users rapidly travel, like pilots navigating spacecraft in the real universe. Users navigate from frame to frame by pointing the mouse cursor at an item linked to another frame and clicking a button. KMS accesses the linked frame and displays it within the same window.



NoteCards

Frank Halasz, Randall Trigg, Thomas Moran
Xerox PARC
1983

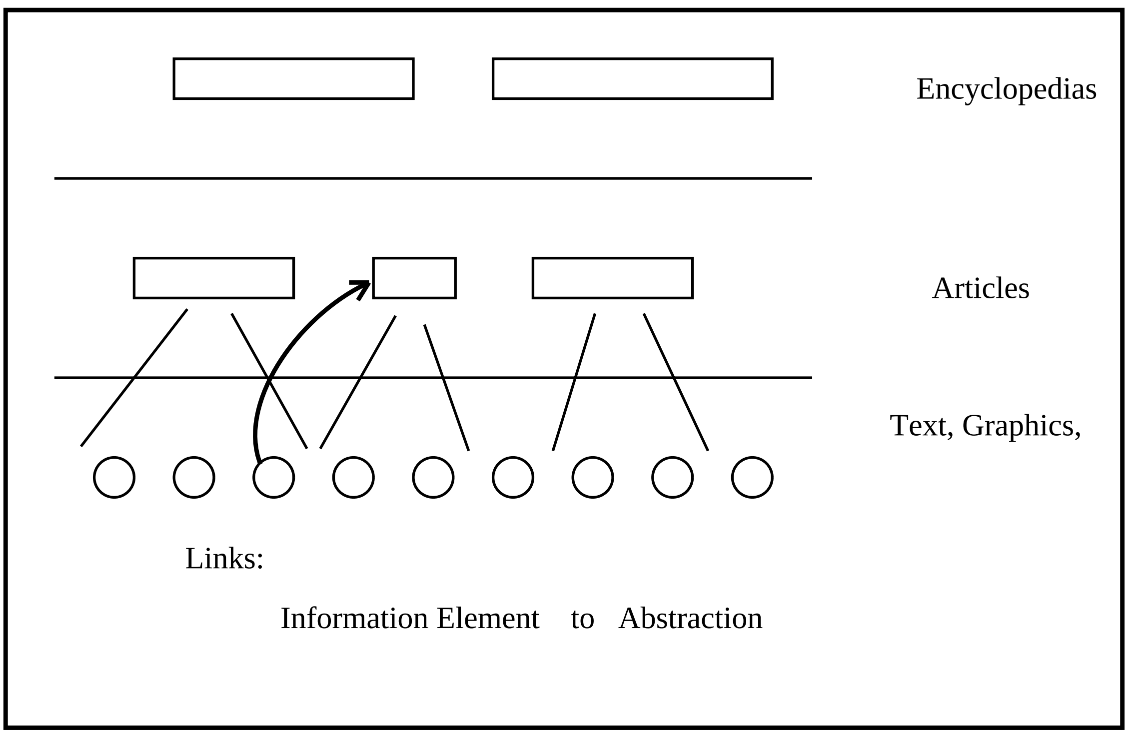
Data model and underlying metaphor. A NoteCard database consists of one or more notefiles containing interlinked notecards. Notecards must be organized into fileboxes within a notefile. A user may link their notecards into arbitrary, non-hierarchical network structures. The central metaphor is an electronic generalization of 3x5 paper notecards which are often used as an idea structuring tool or for information management.



HyperTIES

Ben Shneiderman
University of Maryland
1987

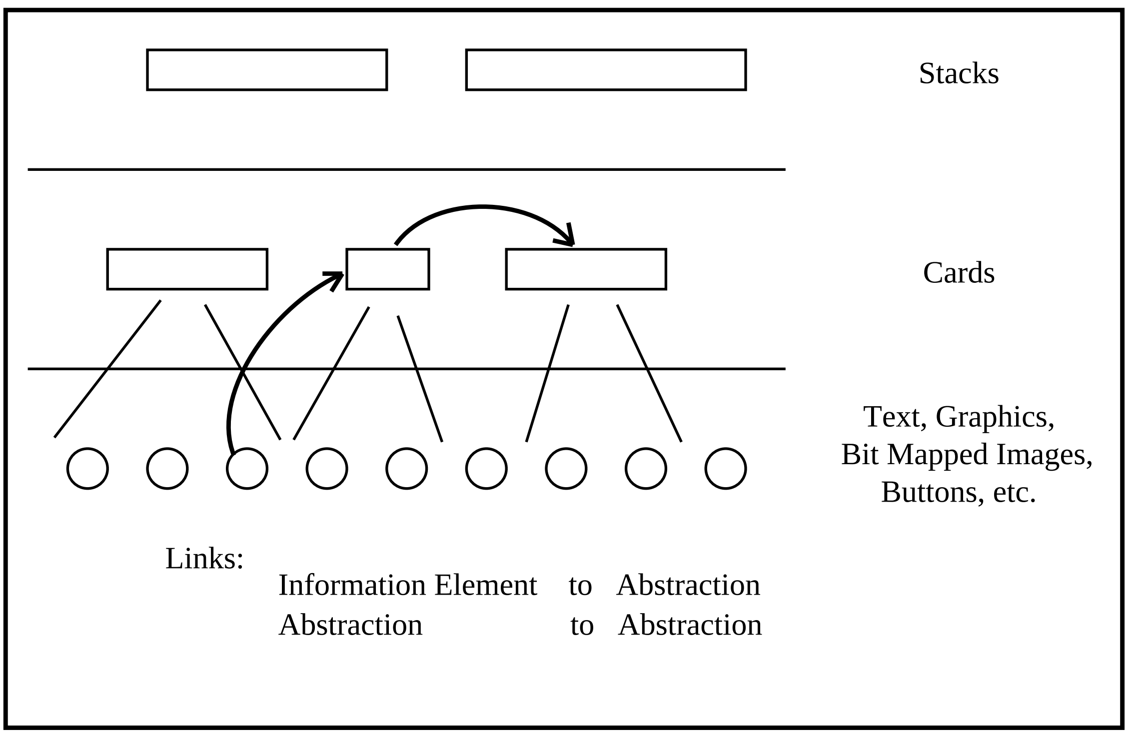
Data model and underlying metaphor. A HyperTIES (*Hyper* The Interactive Encyclopedia System) database consists of a set of arbitrarily interlinked articles. A related set of articles is organized into an encyclopedia. While the resulting structure is commonly a graph, introductory articles may impose a hierarchical organization. The central metaphor is a richly interconnected set of related articles that comprehensively cover a particular topic.



HyperCard

Bill Atkinson
Apple Computer
1987

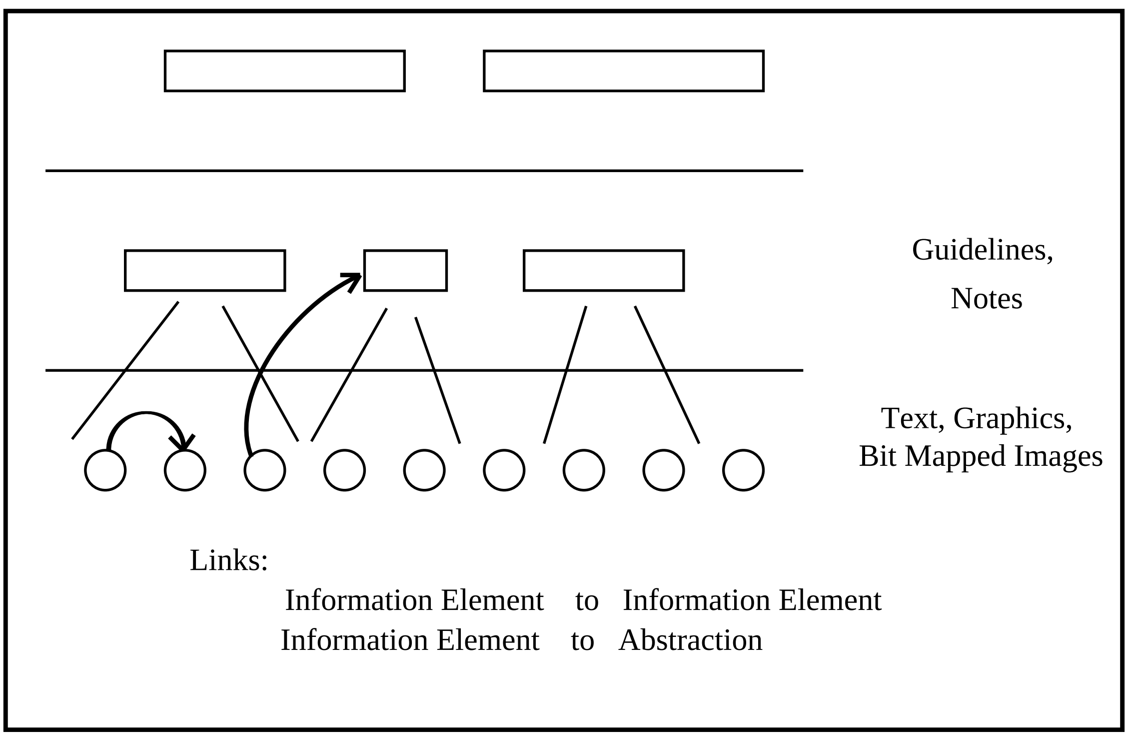
Data model and underlying metaphor. A HyperCard database consists of stacks of cards arbitrarily interlinked by buttons. Stacks may also be used to group cards into hierarchies. The central HyperCard metaphor is the 3x5 paper notecard which is often used as an idea structuring tool or for information management.



Guide

Peter Brown
University of Kent / Owl International, Inc.
1987

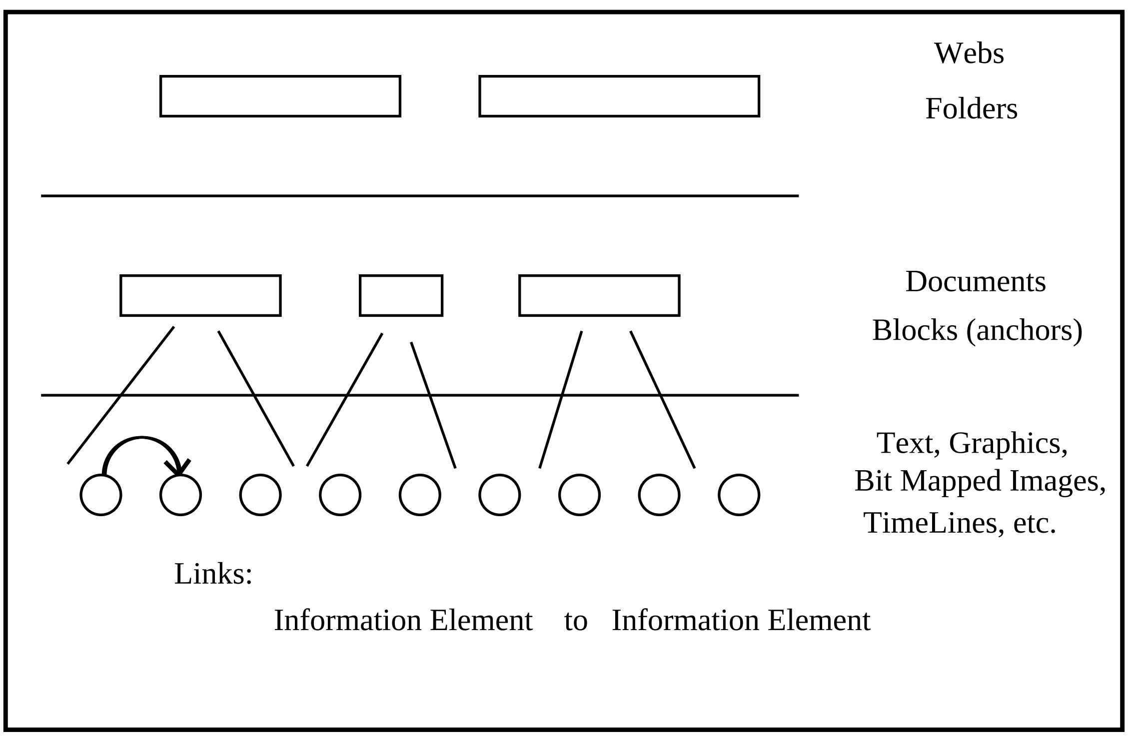
Data model and underlying metaphor. Guide manages a database of guidelines interlinked through buttons. A button is an anchor and its associated link. A Guide database is strongly hierarchical by virtue of its predominant linking mechanism, which implements stretchtext, but arbitrary graph structures may be created. There is no central metaphor to Guide other than it being a tool for electronic "desktop communication."



Intermedia

Norm Meyrowitz
Brown University
1983

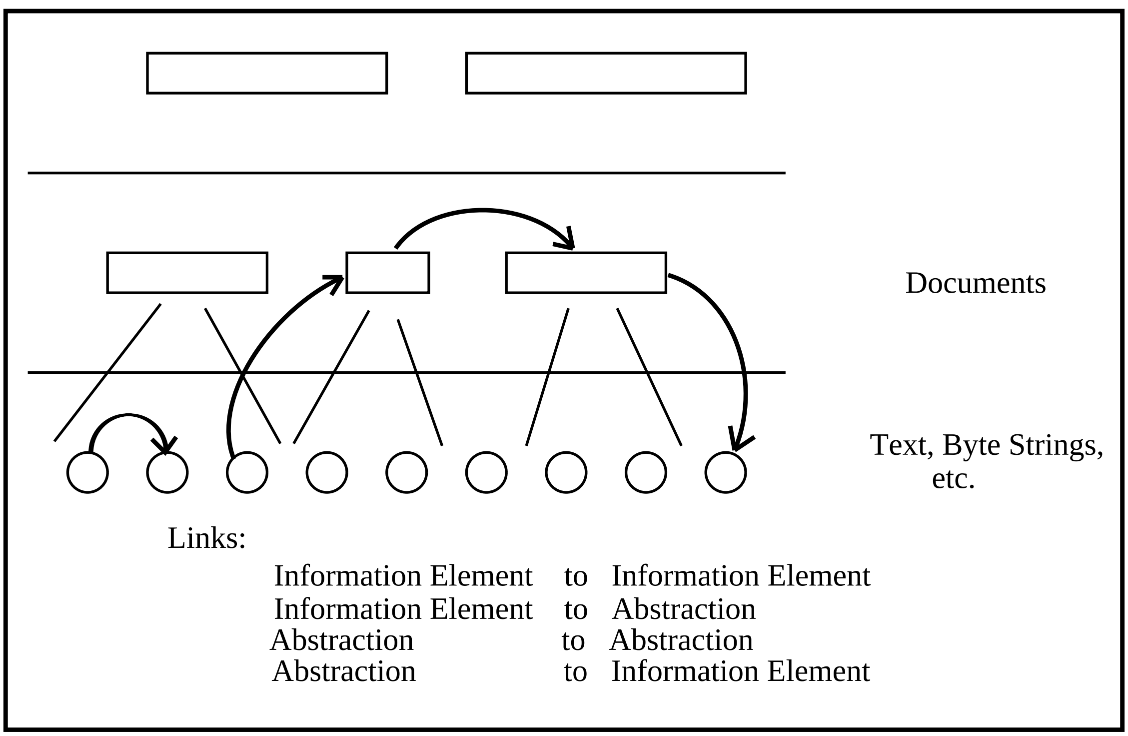
Data model and underlying metaphor. An Intermedia database consists of a set of arbitrarily interlinked documents containing information from a variety of applications. Documents may be organized into folders. A web provides a view on the database of documents. The central metaphor is that of a desktop displaying documents in folders.



Xanadu

Ted Nelson
1969

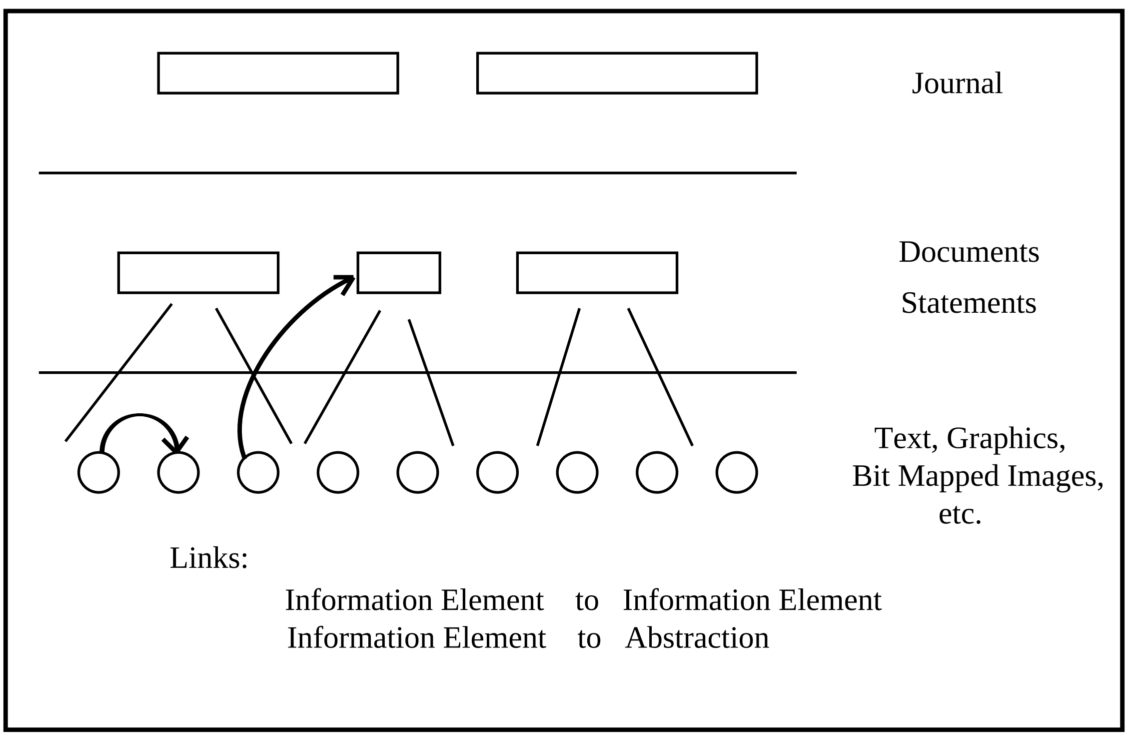
Data model and underlying metaphor. Nodes consist of spans of bytes which have any length and any content. Xanadu is not aware of any logical meaning associated with the information that it handles. The collection of byte-spans and links comprise Xanadu documents.



Augment

Doug Engelbart
Stanford Research Institute
1968

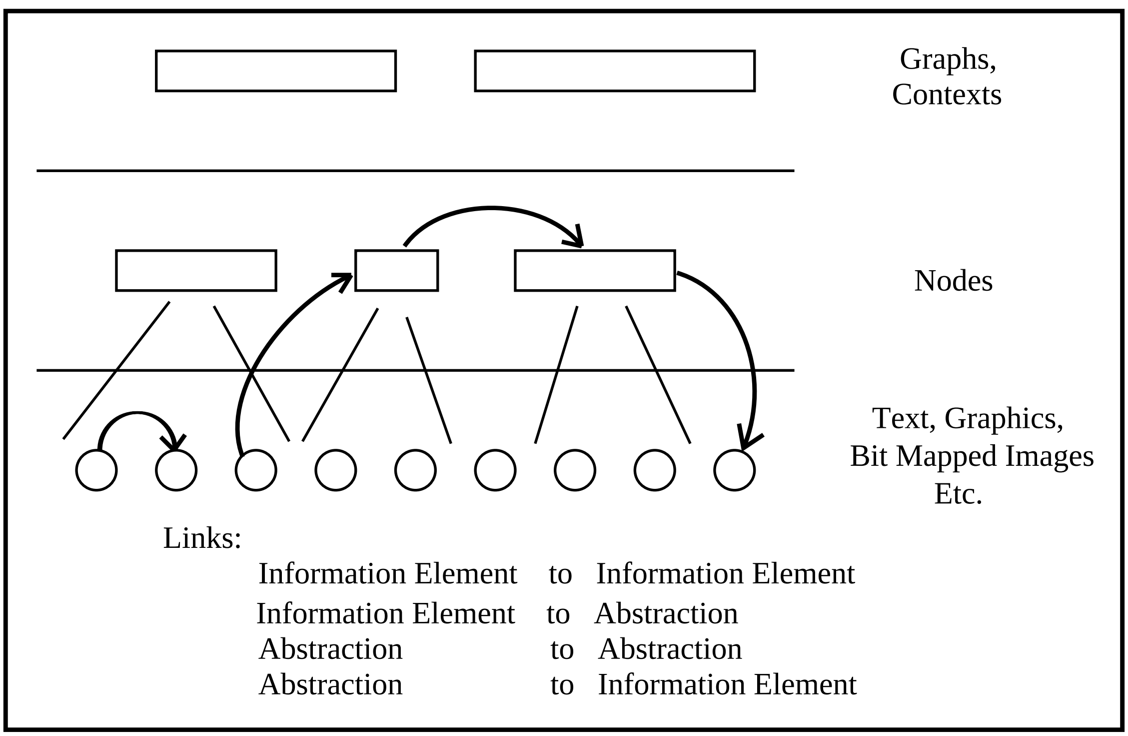
Data model and underlying metaphor. Augment employs explicitly structured files with hierarchically organized nodes, called statements, that contain information. Statements are organized into structures by links; structures are organized into documents. Statements may be referenced individually, so an arbitrary graph structure is possible. The central Augment metaphor is an integrated environment that extends the capabilities of intellectual workers -- the "augmented knowledge workshop."



HAM

Norman Delisle, Mayer Schwartz
Tektronix's Computer Research Laboratory
1984

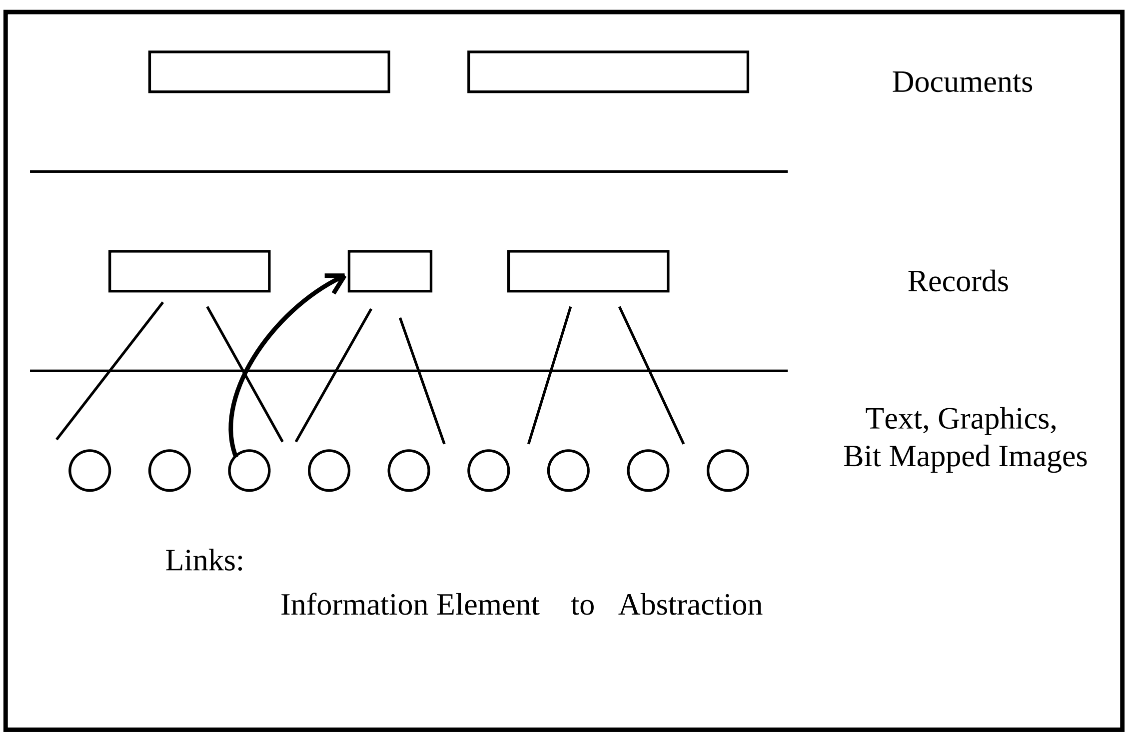
Data model and underlying metaphor. The HAM (Hypertext Abstract Machine) storage model is an arbitrarily-linked, object-based graph structure. The five objects in the system are: graphs, contexts, nodes, links, and attributes. A graph is the highest level object and usually contains all of the contexts relating to a given topic. Contexts partition nodes within a graph. Nodes contain information. Arbitrarily forged, bi-directional links define relationships between nodes. Attribute/value pairs attached to contexts, nodes, or links give semantics to HAM objects and can represent application-specific properties of objects or contain information that further describes an object. There is no central metaphor to HAM.



Concordia / Document Examiner

Janet Walker
Symbolics, Inc.
1983

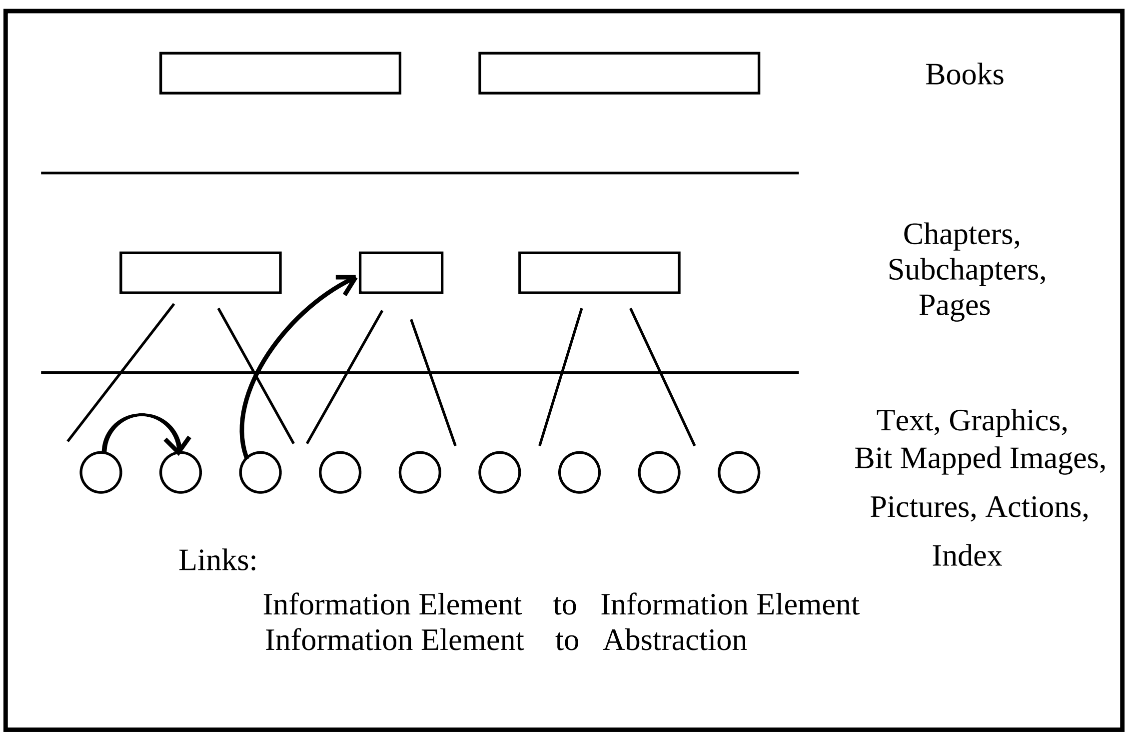
Data model and underlying metaphor. A Concordia/Document Examiner database consists of structured records linked together to form a document. A record is the smallest unit of information accessible from Document Examiner and is the basic building block from which writers construct entire documents. The final organization of a document is a result of the structure imposed by the way records are linked. Since the system is designed for publishing books, the data model tends to be strongly hierarchical; however, arbitrary graph structures may also be formed. There is no central metaphor to Concordia/Document Examiner.



IGD

Steve Feiner, Andries van Dam
Brown University
1982

Data model and underlying metaphor. An IGD (Interactive Graphical Document) database consists of a network of full-screen sized pages connected by links. Pages can be nested arbitrarily deep in a hierarchy of chapters. Chapters may be organized into books. The central metaphor is that of a dynamic electronic book.



An application-oriented taxonomy of hypertext systems

Any taxonomy of Hypertext systems is somewhat arbitrary since a particular system may be used (or misused) for many purposes ...

Conklin:
- Macro Literary Systems
- Problem Exploration Tools
- Browsing Systems
- General Hypertext Technology

Leggett:
- Literary
- Structural
- Presentational
- Collaborative
- Explorative

Literary Hypertext Systems

Characteristics
- Links are more important
- Smaller granularity of link referents
- Information element to information element associations
- Free annotation

Example applications
- Reading, writing, publishing, critiquing
- Education, scholarship

Example systems
- MEMEX, NLS/Augment, Xanadu, Textnet, Intermedia, Neptune/HAM, IGD, Guide

Structural Hypertext Systems

- Characteristics
  - Nodes are more important
  - Larger granularity of link referents
  - No information element to information element associations
  - No (or very restricted) annotation

- Example applications
  - Information management
  - Argumentation
  - Idea processing

- Example systems
  - KMS, gIBIS, NoteCards, HyperCard

Presentational Hypertext Systems

Characteristics
- Usually Structural Hypertext Systems
- Separate author and browser

Example applications
- Information kiosks
- Reference material - dictionaries, encyclopedias
- Documentation

Example systems
- HyperTIES, Concordia/Document Examiner

Collaborative Hypertext Systems

- Characteristics
  - Links and nodes are important
  - Information element to information element associations
  - Free annotation
  - Distributed
  - Security

- Example applications
  - Software engineering environments
  - Organizational information management

- Example systems (Limited)
  - Neptune/HAM, NLS/Augment, Intermedia, KMS, NoteCards

Explorative Hypertext Systems

Characteristics
- Links and nodes are important
- Information element to information element associations
- Free annotation
- Space

Example applications
- Brainstorming
- Early activities of writing, thinking
- Problem formulation and exploration

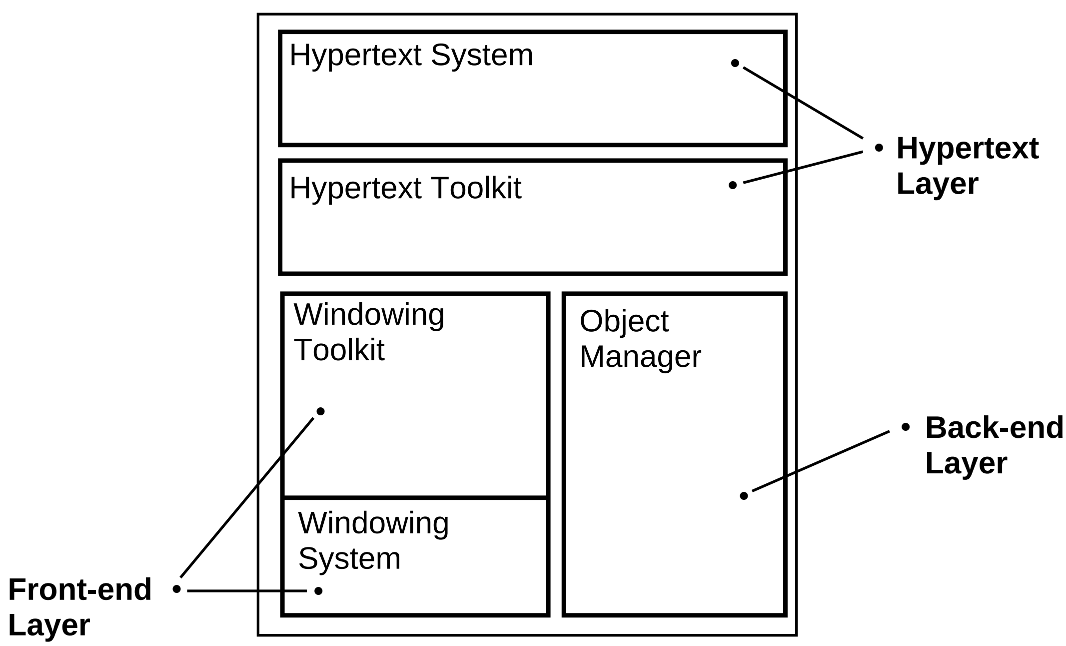
Example systems (Limited)
- KMS, NoteCards

General architecture

A simple layered architecture is proposed as a conceptual framework for organizing discussions of hypertext system functionality and design. As shown below, a hypertext system is assumed to be composed of three layers of functionality: front-end, hypertext, and back-end layers. Each function present in a system may be categorized as being a front-end, hypertext, or back-end function.

A hypertext system front-end may use functionality present in a windowing system or windowing toolkit and typically includes display operations and capturing user input. Back-end functions are provided by an object manager and include retrieving, storing, and caching. The hypertext layer implements the functionality and data abstractions required to translate user requests at the front-end into actions in the back-end. Hypertext layer functionality may be further separated into hypertext toolkit and hypertext system layers.

In most cases, one would expect functions in the various categories to be implemented in their respective layers. However, boundaries between the three layers are not fixed, and in the design of a hypertext system, there may be an option of implementing a function in either of two layers. For a more complete discussion of hypertext system functionality see Schnase *et al.* [1988] and Kacmar *et al.* [1988].



Where do hypertext systems fit?

What is Hypertext?
- Back-end view
  - *Hypertext is an object-oriented database with retrieval by browsing instead of query ...*
  - *Hypertext is a spatially-oriented database with retrieval by navigation instead of query ...*
- Data structures view
  - *Hypertext is an arbitrary graph (with corresponding complex algorithms*) ...
- Front-end view
  - *Hypertext is a direct manipulation user-interface to an information management system ...*
  - *Hypertext allows the user to directly manipulate information and structure ...*
- AI view
  - *Hypertext is a semantic network ...*
  - *Hypertext is an informal knowledge representation scheme ...*
- Literary view
  - *Hypertext is a new medium for communication ...*

Where do Hypertext systems fit?
- Depends on view, intended use, and understanding

Opinion
- Must be viewed as a system -- a complex system
- Close relatives are operating systems and database systems
- Hypertext systems have stringent requirements on:
  - User-interface
  - Speed of retrieval
  - etc.

Information development

Linear to non-linear conversion: A case study

Automatic Conversion of Linear,
Paper-Based Documents into Hypertext

The Rexx Project

Dr. John Leggett
Hypertext Research Lab
Department of Computer Science
Texas A&M University
1987

Introduction
Rexx Project Overview
Description of Rexx Manual
Description of HyperTIES
Manual Conversion
Automatic Conversion
Topology of Rexx HyperTIES Encyclopedia
Lessons Learned from the Rexx Project
Reflections on the Rexx Project
Current and Future Work

Introduction

Most information is currently in linear, paper-based form

Cost of producing, distributing and maintaining is becoming prohibitive

Medium is changing from paper to electronic

Paper may not be an option ... Space Station Documentation

We are no longer bound by the linear format

Hypertext is a non-linear medium

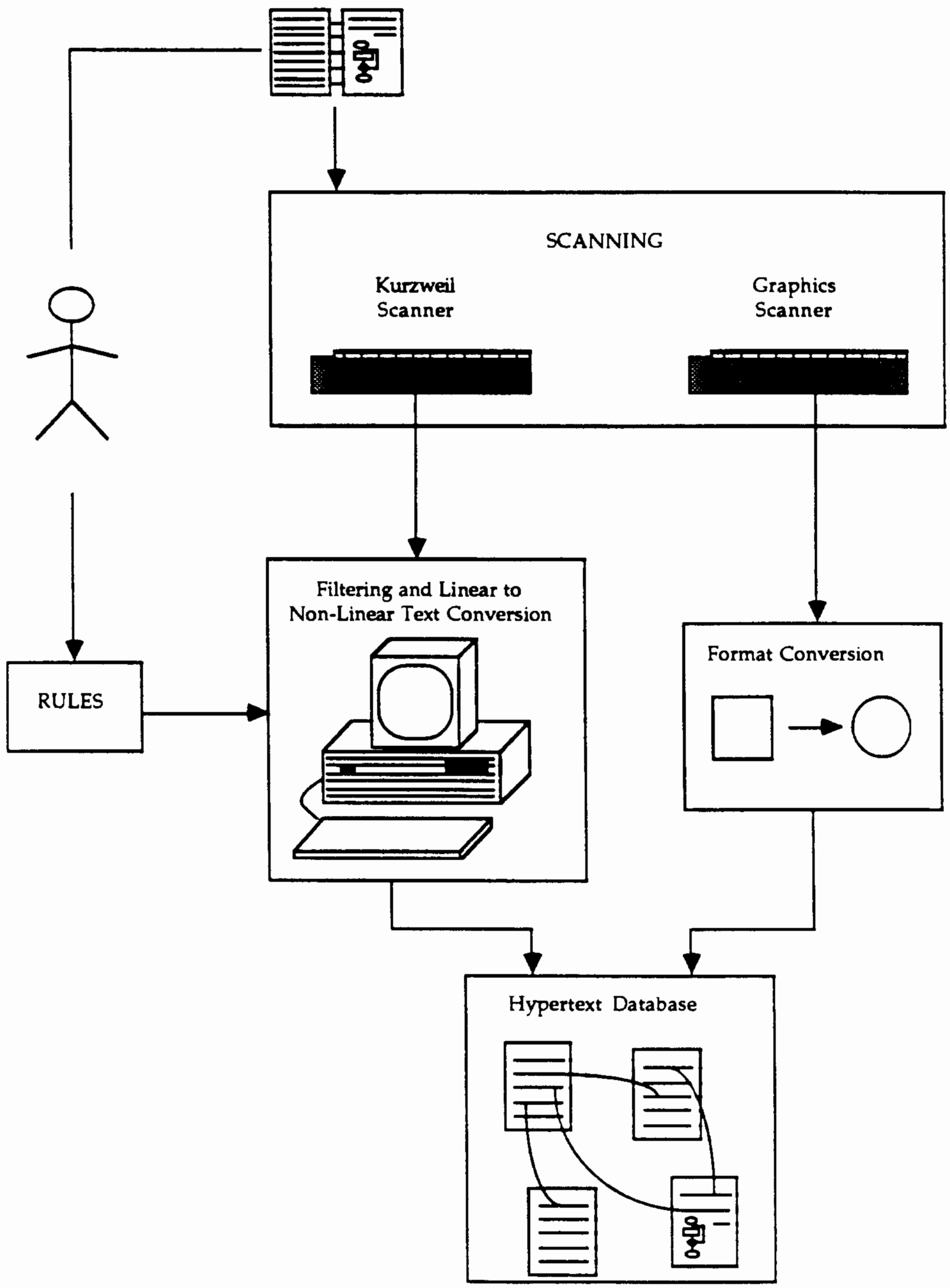
Authors will learn to write non-linearly

Many dollars are invested in current documentation

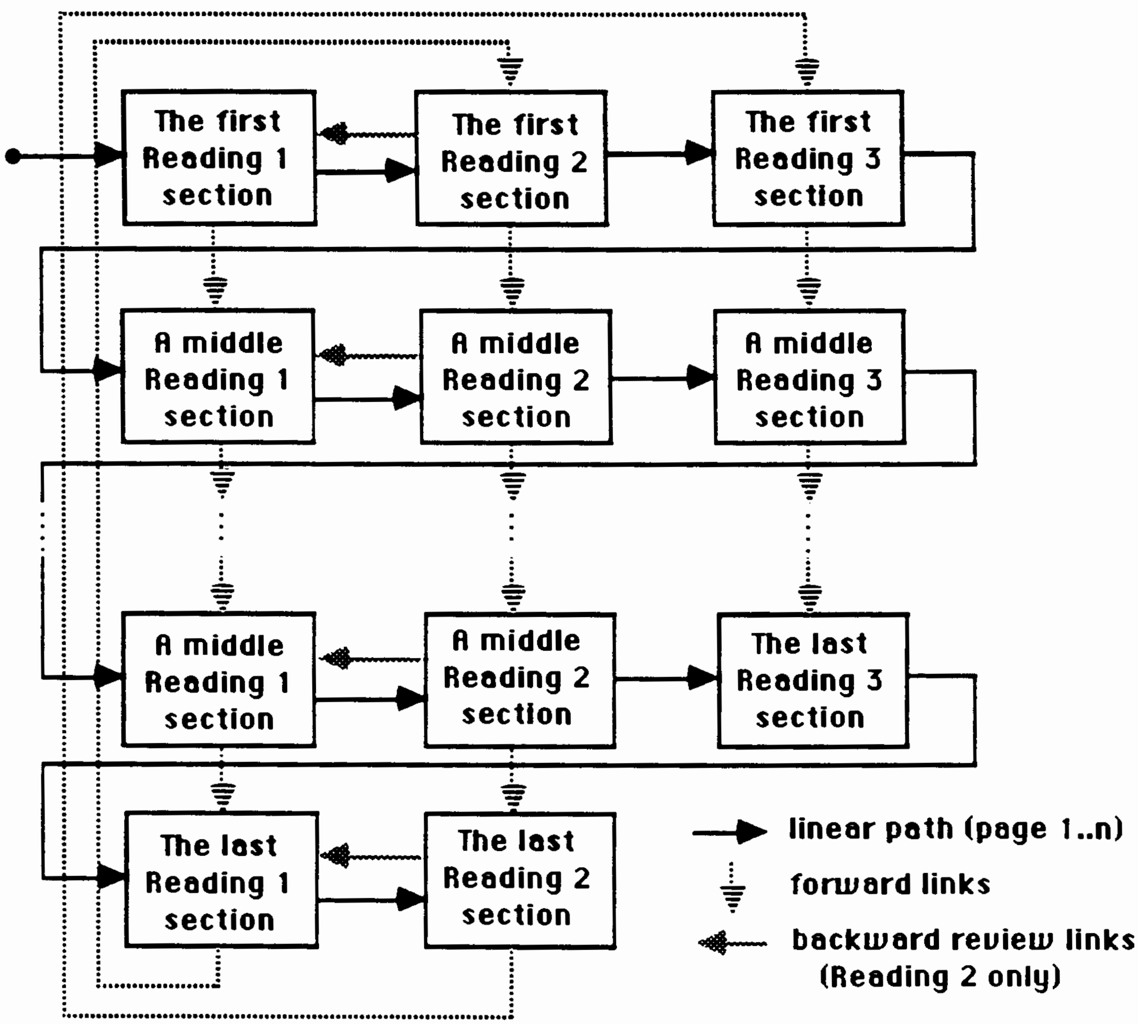
To speed the conversion of massive amounts of linear documentation to non-linear form, we must have an automatic process

The Rexx project was a case study intended to shed some light on the process and associated problems of converting from a linear, paper-based form into hypertext ...

Rexx Project Overview



Description of Rexx Manual



Description of HyperTIES

Hyper - The Interactive Encyclopedia System

Univ. of Maryland (Ben Shneiderman)

Presentational hypertext systems - "write once, read-mostly" environments

Author and Browser are separate

Nodes: textual (articles) and bit-mapped images (pictures)

Links: uni-directional, information element to abstraction links

Display: small, tiled, paged windows

Platform: IBM PCs

Limitations: maximum of 200 articles per encyclopedia
maximum of 10,500 characters per article

HyperTIES was chosen as the target hypertext system because of its availability, simple storage design and separate Author and Browser. The output of the automatic conversion program is directly browsable by the HyperTIES Browser.

Manual Conversion

Why? To generate the rule base used in the automatic conversion

How? Pasted to wall, colored lines, 7 hours

Observations

Negatives:

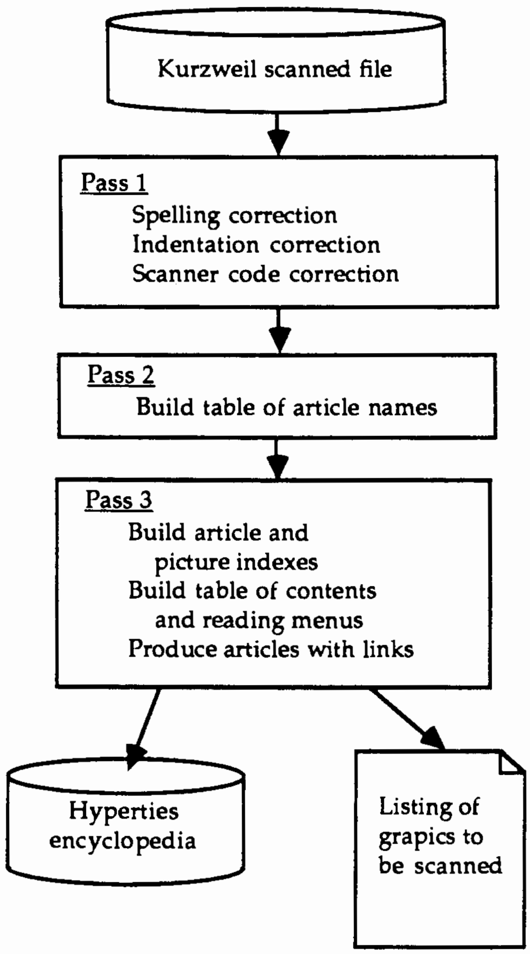
- Limit on number of articles forced larger granularity (<200 articles)
- Limit on size of articles (<10,500 characters) forced articles to be split
- Larger granularity requires into-links
- Approximately 500 articles would be necessary if we only used to-links
- A large amount of text in the paper version is irrelevant in a hypertext environment: headers, footers, paper-link instructions, etc.

Positives:

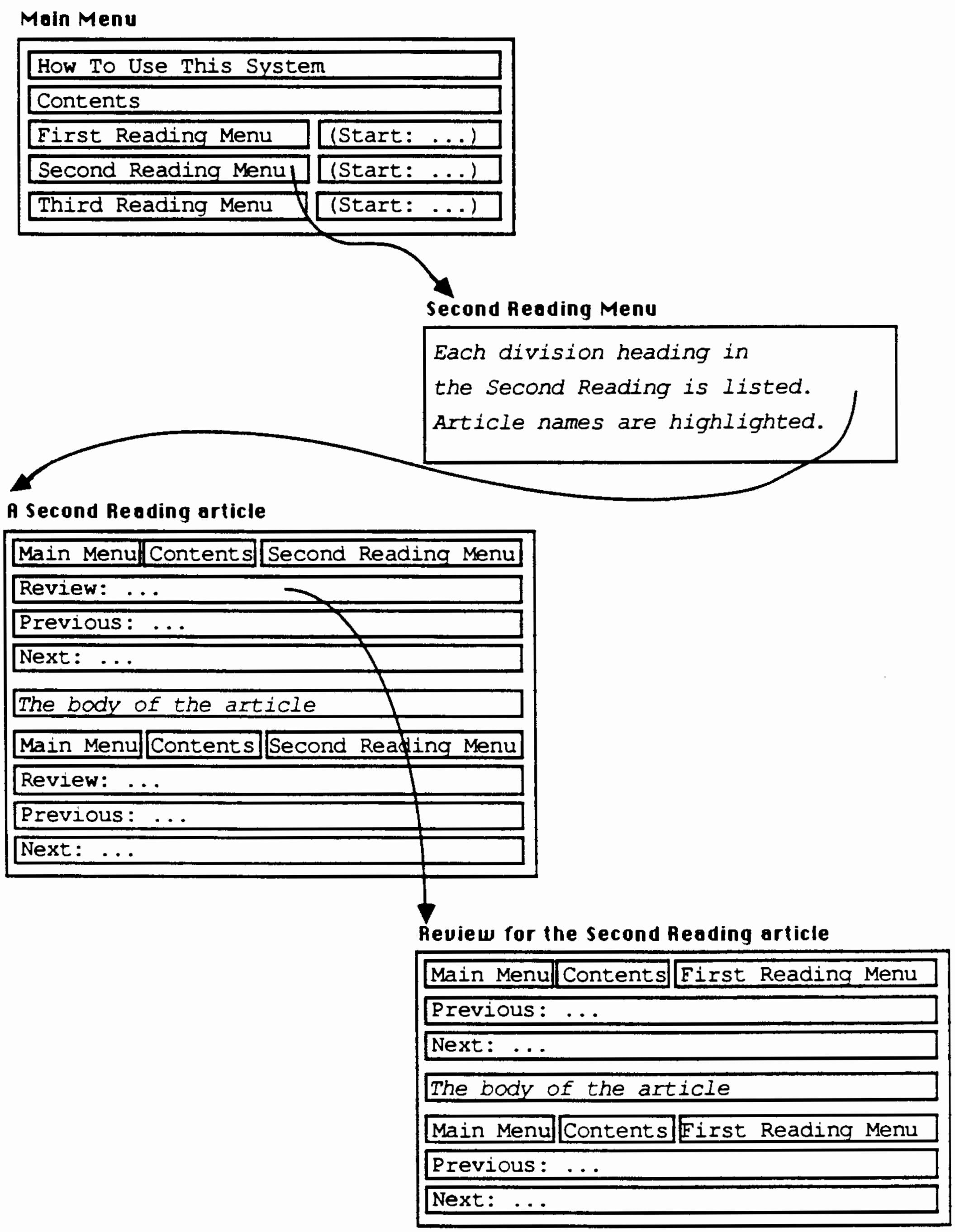
- Keyword template matching would work quite well since the manual was written in a regular fashion

Why did we continue? To see what else could be learned and to verify that a simple keyword template matching algorithm would work

Automatic Conversion



Topology of Rexx HyperTIES Encyclopedia



Lessons Learned from the Rexx Project

Negatives:

- Scanning with Kurzweil 4000
  - Operator is **very** important
  - Ligatures and kerning
  - Separate graphics scanning
  - Summary: You don’t want to do it! (commercial services)

- Rexx manual was probably not a good choice
  - Highly piece-wise linear
  - Already chunked into "small" units

- HyperTIES was probably not a good choice
  - PC/AT platform too slow
  - Display too small (single, paged window)
  - No into-links
  - Limited graphics

Positives:

- Keyword template matching worked well for this level of conversion

Reflections on the Rexx Project

Granularity of node size
- Inherent to hypertext construction and hypertext systems in general
- Inherently tied to display techniques

Into links are a necessity
- Consider keyword index
- Consider granularity of node size

We forged only explicit structural and explicit associational links producing a "linear hypertext" ... i.e., a non-linear text that may be browsed in a non-sequential manner but is almost entirely devoid of higher level semantic links

Since the information has already been "chunked" for paper, it may have to be "re-chunked" for hypertext

Current and Future Work

Conversion of a set of manuals into an integrated hypertext
- Redundancy
- Appropriate views

Forging implicit associational links
- Natural language processing
- Knowledge-based techniques (causal modeling)
- Usage-driven dynamic links

Considerations for creating hypertext

- Inter-Document Considerations
- Intra-Document Considerations
- Structures and Associations
- Context
- Questions
- Issues
- Other Conversion Considerations

Inter-Document Considerations

Redundancy
- Verbatim - exact or nearly exact repetition
- Granular - same material at different level of detail

Views
- Each document has its own view, style and intended audience
- Should the same views be retained in the integrated hypertext?
- What other views are appropriate?

Inter-Document links
- Degree of connectivity

Intra-Document Considerations

A priori assumptions about the reader's path

Previous references

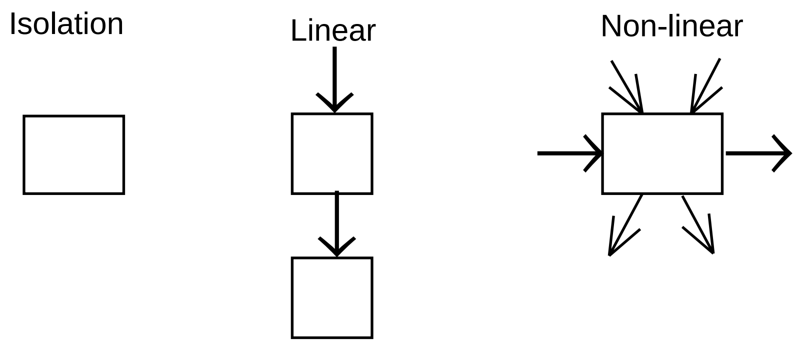
Intra-Document links
  Degree of self-referencing

Chunks that may be read in isolation

Chunks that may be read in many different paths

Structures and Associations

Structures



Feature extraction: books, chapters, sections, tables of contents, indexes

Explicit associations
e.g. See X in Y
Function of writer's style, inclination, knowledge of subject matter
Environment of document

Implicit associations
Higher level semantic associations
Full text cross references and proximity searches
Natural language understanding, linguistic techniques
Usage-based dynamic linking

Context

Writing is always done in context

It is a social context most clearly seen in collaboration

How can we write isolated, context-less chunks?

Q. Do previous references have to be removed?
A. Not as long as we can distinguish the appropriate context

Q. How can we distinguish the appropriate context?
A. Threads: history, main, alternatives

Q. Does this imply precedence hierarchies and rules for linking?
e.g. If you link to me then you must have a definition of *x* in your "precedence hierarchy" ...

Questions

How reusable are the modules of text?

Will the modules fit into different rhetorical contexts?
Different: approaches, background, reading level?

Will document quality diminish through reuse of text modules?

When is hypertext not appropriate?

Issues

At a minimum, a hypertext document must be as usable as a paper document ... this will be extremely difficult to achieve

A hypertext document must:
- Allow free annotation
- Allow personalization
- Have the concept of space (to support the above)
- Support multiple views

A poor linear document may be even worse when non-linearized

We must understand the various forms of on-line information and the nature of the users of that information

Other conversion considerations

Personality of writer

The information has already been "chunked" two ways:
- Logically for consumption by the reader
- Physically by any formatting commands and mark-up

Rhetoric of hypertext ("What are we doing, anyway?")

Preliminaries

Rhetoric of hypertext

What must we understand to do it right?

We should probably rewrite

Preliminaries

What is hypertext?

"Non-linear writing" and

" ... a body of written or pictoral material interconnected in such a complex way that it could not conveniently be presented or represented on paper." [Ted Nelson 1965a]

Bush vs. Nelson

Trails over the public record (which is linear) with associations by user and trailblazer	*vs*	non-sequential writing with associations by user

On-line documentation vs. hypertext

Q. If you have created a hypertext in a "hypertext" system whose major emphasis is paper ... have you created a hypertext?

We currently have an inadequate conceptual model upon which to build text to hypertext conversion programs

Rhetoric of hypertext

Rhetoric: the theory and practice of effective communication

We don't even know how to display linear sequences so that they are easily read

Style always begins with the medium, and we currently don't understand the medium

We have no rhetoric for hypertext ... how can we write/translate effectively?

Composing in hyperspace requires the concurrent design of nodes, links and environment.... this is very difficult

We must understand how to read hypertext before we can write hypertext for the readers

I don't doubt that you can create/author very good specific hypertexts for training, CAI, museums, etc. However, so far they are static, one-time shots which take great time and great expense...

What must we understand to do it right?

The various styles of documentation

The various styles of hypertext

How to read non-linearly

How to write non-linearly

How to display non-linear writing of various styles

The concept of space
- Coffee stains on well-used pages, personalization, annotations ...
- Old copies of manuals kept for years due to annotations ...

We should probably rewrite

"We live in the rear-view mirror." [Marshall McLuhan 1969]

We use new technology to do what we did yesterday ...

Not negative ... just cause to stop and think and not let our paper model cramp hypertext

This medium can be so much more than paper!

Information presentation

User-interface requirements

Must be fast ... speed is critical

Should provide direct manipulation of information and structure

Must be able to display and manipulate: nodes, links, anchors
and their names and types

Should provide a simple conceptual model
which does not conflict with the hypertext model

Should be tailorable

Should be extensible

Windowing options
- Tiled vs. overlapping
- Scrollable vs. non-scrollable
- Spatial vs. non-spatial
- Number of windows
- Power vs. complexity

Issues
- How do you show links and their scope?
- How can you give an indication of the contents of a link’s destination?
- Is the desktop metaphor appropriate for hypertext?
- Can the hypertext metaphor suffice for all interface requirements?

Browsing strategies

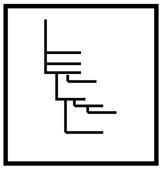
- Navigation is the primary information retrieval method
  - Speed is critical
  - A simple conceptual model is critical
  - Orienting cues are important
    - Bookmarks
    - Common indices
    - Free annotation for marginalia
    - Histories with path clearly indicated
    - Trails
  - Cognitive overhead and disorientation are major problems

- Information browsers
  - Granularity of information access
  - General information browsers
  - Application specific browsers
  - Power vs. complexity

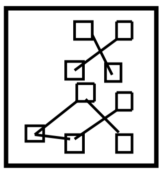
Browsing strategies (cont.)

Structural browsers

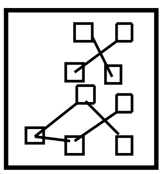
Outline browsers

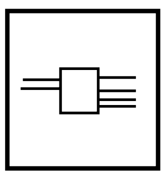


Graphical browser



Global vs. local view





Capabilities

- Overview
- Manipulate structure
- Manipulate information

When are structural browsers appropriate?

Problems with the hypertext model

Cognitive overhead

Mental effort required to hold the hypertext model and
your position in that model's world

Authoring
- Mental effort required to create, name, type and maintain links
- Problems
  - Idea capturing with little overhead
  - Deciding how/where to link new material into the network

Reading
- Mental effort required to decide which of several links to traverse -- then remembering your traversal strategy ...

Solutions
- Concept of space
- Conceptually simple ("thin") interface
- Speed
- Allow information elements to be first class objects (i.e. exist "on their own" or "outside of any abstraction")
- Allow for "description" of destination node's content to be displayed without actually traversing the link (e.g. pop-up descriptions, etc.)

Disorientation

Getting "Lost in Hyperspace" ...

Information space has no natural topology ...

Questions to be answered:
- Where am I? How did I get here? How do I get back?
- How do I get to another place?
- Where is *X* exactly? Where is *X* approximately?
- What is the topology of the current information space?

Linear vs. non-linear
- Physical cues with paper
  - Visual: color, size, shape
  - Weight
  - Wear and tear
- Physical cues for displays
  - Visual: color, size, shape

Solutions
- Concept of space
- Concept of time
- Use of hierarchies
- Personalization
- Graphical browsers
- Filtering: types, attribute/value pairs, time, space
- Histories
- Trails
- Search and query: keyword, types, attribute/value pairs, time, space

Summary of research labs and projects

Who: Jeff Conklin, Mike Begeman

Where: MCC/STP

What: Planetext, gIBIS, NLC

Research on the use of Hypertext in software engineering environments

Who: Mayer Schwartz, Norm Delisle

Where: Computer Research Lab, Tektronix Labs in Beaverton, Oregon

What: HAM, Neptune

Research in Hypertext systems / back-ends; formalization

Who: Frank Halasz, Randy Trigg

Where: Xerox PARC (Aarhus University)

What: NoteCards, Textnet (Ph. D.), CoLab

Research in collaborative use of Hypertext

Summary of research labs and projects (cont.)

Who: Jan Walker

Where: DEC / Cambridge Research Lab (CRL)

What: Symbolics Concordia / Document Examiner

Research in computer-based documents and document engineering

Who: Apple (Tim Oren, Steve Weyer, ...)

Where: Cupertino, California

What: HyperCard

Who: Sun (Amy Pearl, Reg Gillmor)

Where: California, Massachusetts

What: Sun Link Services, Sun 386i OnLine Help

Summary of research labs and projects (cont.)

Who:	Knowledge Systems, Inc. (Rob Akscyn, Don McCracken)
Where:	Murrysville, Pennsylvania
What:	Knowledge Management System (KMS)

Who:	Roger Gregory (Ted Nelson)
Where:	AutoDesk, Inc.
What:	Xanadu, AutoCad with Hypertext capabilities

Who:	Norm Meyrowitz
Where:	IRIS (Institute for Research in Information and Scholarship) at Brown University
What:	Intermedia
	Research in Hypertext systems and their use in higher education

Summary of research labs and projects (cont.)

Who: Steve Feiner

Where: Columbia University

What: Interactive Graphical Document (IGD) Project

Research in computer-based documents

Who: John Smith

Where: University of North Carolina

What: Writing Environment (WE)

Research in computer-based writing tools

Who: Ben Shneiderman

Where: University of Maryland

What: HyperTIES

Research on presentational (encylopedic) Hypertext systems
Conversion of linear documents to non-linear form

Summary of research labs and projects (cont.)

Who:	Walt Scacchi
Where:	University of Southern California
What:	Document Integration Facility (DIF)
	Research on the use of Hypertext in software engineering environments

Who:	Doug Engelbart
Where:	Stanford
What:	NLS / Augment
	Research on augmenting the human intellect and collaborative systems

Who:	Peter Brown
Where:	University of Kent at Canterbury
What:	Guide
	Research on on-line help systems

Summary of research labs and projects (cont.)

Who: Gregory Crane

Where: Harvard

What: Persius Project

Building a large Hypertext of the classical Greek world for future use in scholarship and higher education

Who: Mark Frisse

Where: Washington University School of Medicine

What: Hypertext Medical Handbook

Research on computer-based documents

Who: Bruce Schatz, Michael Caplinger

Where: Bell Communications Research

What: Telesophy

Research on a universal library of all information accessed at a distance

Summary of research labs and projects (cont.)

Who: John Leggett, Craig Boyle, John Schnase, Chuck Kacmar

Where: Texas A&M University

What: Hypertext Research Lab

Research in Hypertext systems
Design of next generation Hypertext systems
Conversion of linear documents to non-linear form

Research and design issues

Nodes

Granularity

Typing

Different applications for different types?

Naming
  User vs. system

Concept of space

Object-based information elements?

Composites

Links

Uni-directional, bi-directional, or strap

Granularity of link referent

Explicit vs. implicit addresses

Typing

Naming
- User vs. system

Static vs. dynamic (computed)

Include vs. reference semantics

Organizational (structural vs. associational)
- External
- Internal
  - Specifiers
  - Are all nodes composites?

Embedded vs. non-embedded

Ensuring integrity
- "Back" links

Anchors

Multiple link referents

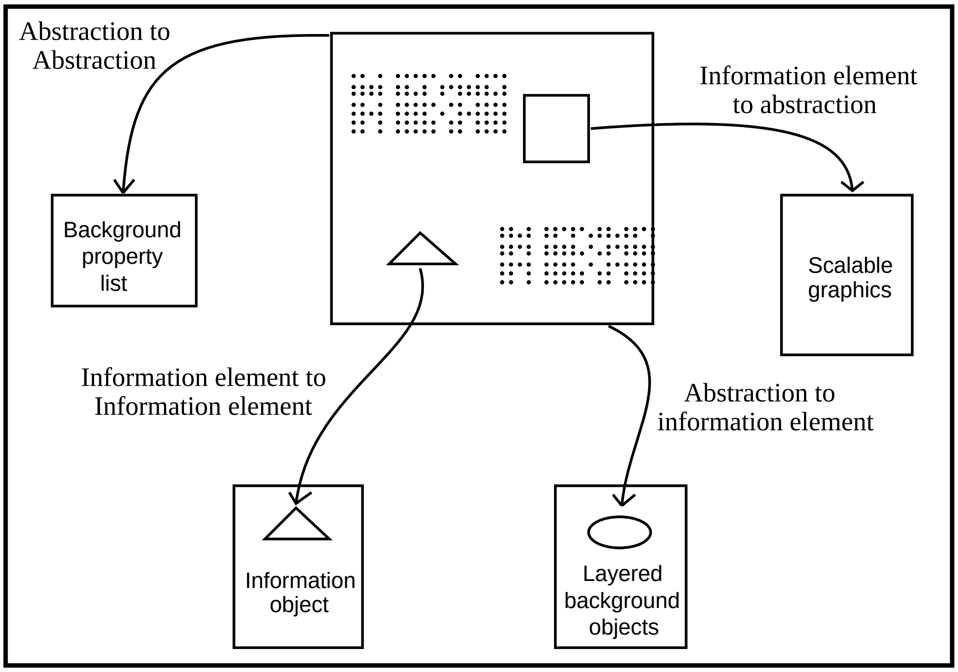
Viewspecs

Composite/version identification

Are nodes and links different or is it all in the interface?

Composition

Inclusion semantics for links



Structural
- Include specifier
- Object resides in one composite

Associational
- Links with include semantics
- Object may reside in multiple composites
- Link must be to object in particular composite (recursive)
  - Anchors may hold this information in addition to viewspec
- Constraints
- Versions (see next page)

Naming of objects is a critical issue
- Unique object identifiers
- Global and local parts to identifiers
  - Global part - unique in "world"
  - Local part - composite (context, version) specifier

Versioning

Version history trees

Automatic vs. on-demand

Time vs. object

Granularity
- Object, node, tree, group, web, document, etc.
- Feature (set of coordinated changes)

Forward / backward deltas

Regeneration / reconstitution
- Which versions?
- Caching of particular versions

Versions and deltas are important

Composite versioning
- When versioning a composite object:
  - Do you version the composite?
  - Do all composites reflect the new version?   Is it selectable?

- When versioning a composite:
  - Do the composite objects get versioned?

Collaboration and distribution

Collaboration requires shared access to a distributed database

Concurrency control
- Locking
- Granularity

Security / protection
- Simple read/write is not appropriate
- Example rights
  - Existence
  - No access
  - Read only
  - Annotate only
  - Link only
  - Write access
  - Full access (delete, etc.)
- Granularity

Shared vs. private contexts
- Merging private contexts into shared contexts

Events maintenance
- Access events
- Update events

Search and query

Navigation is not sufficient in general ...
- Depends on richness of network structure

Content search
- Information retrieval techniques (keyword, index, full-text, etc.)
- Nodes, links, attribute/value pairs
- Filtering
- Scope (contexts, webs)

Structure search
- Totally misses the associations!
- Structure query languages (Mendelzon)
- Restricted searching is necessary due to complexity of graph search

Spatial searching
- Contents of nodes
- Hyperspace

Temporal searching

Computation and virtual structures

Static vs. dynamic (virtual) components
- Changing information and/or structure ...

Dynamic nodes, links, composites
- Existence and/or contents is decided at node retrieval time
- Requires search and query mechanism or computational facility
- If....Then....Else clauses
- Thresholds

Computational engines
- Externally through programmer's interface
- Internally through integrated extension language

Examples
- Summary nodes
- Automatic trail blazing
- Specific applications

References

Suggested Hypertext References

Akscyn, R., McCracken D., and Yoder, E. 1988. KMS: A distributed hypermedia system for managing knowledge in organizations. *Commun. ACM*, 31, 7, (July), 820-835.

Bush, V. 1945. As we may think. *Atlantic Monthly*, 176, 1, (July), 101-108.

Caplinger, M. 1987. An information system based on distributed objects. *Proceedings of the OOPSLA '87 Conference*, (October), pp. 126-137.

Conklin, J. 1987. A survey of hypertext. MCC Technical Report No. STP-356-86, Revision 2, Microelectronics and Computer Technology Corporation, Austin, TX.

Delisle, N., and Schwartz, M. 1986. Neptune: A hypertext system for CAD applications. *Proceedings of the ACM International Conference on the Management of Data (SIGMOD)*, pp. 132-143.

Delisle, N., and Schwartz, M. 1987. Contexts – A partitioning concept for hypertext. *ACM Trans. Off. Inf. Syst.*, 5, 2, (April), 168-186.

Engelbart, D. C. 1963. A conceptual framework for the augmentation of man's intellect. In *Vistas in Information Handling*, Howerton and Weeks, Eds., Spartan Books, Washington, DC, pp. 1-29.

Engelbart, D. C. 1984. Authorship provisions in AUGMENT. *Proceedings of the 1984 COMPCON Conference*, *COMPCON '84 Digest*, (San Francisco, Calif., February), IEEE Computer Society Press, San Francisco, Calif., pp. 465-472.

Engelbart, D. C. 1984. Collaboration support provisions in AUGMENT. *Proceedings of the 1984 AFIPS Office Automation Conference, OAC '84 Digest* (Los Angeles, Calif., February), pp. 51-58.

Feiner, S., Nagy, S., and van Dam, A. 1982. An experimental system for creating and presenting interactive graphical documents. *ACM Trans. Graphics*, 1, 1, (January), 59-77.

Garg, P. K., and Scacchi, W. 1987. On designing intelligent hypertext systems for information management in software engineering. *Proceedings of the Hypertext '87 Conference*, (Chapel Hill, NC, November), pp. 409-432.

Halasz, F., Moran, T., and Trigg, R. 1987. NoteCards in a nutshell. *Proceedings of the CHI '87 Conference on Human Factors in Computing Systems*, (Toronto, April), J. M. Carroll and P. P. Tanner, Eds., ACM, New York, pp. 45-52.

Halasz, F. 1988. Reflections on NoteCards: Seven issues for the next generation of hypermedia systems. *Commun. ACM*, 31, 7, (July), 836-852.

Kay, A. C., and Goldberg, A. 1977. Personal dynamic media. *Computer*, 10, 3, (March), 31-41.

Landow, George. 1987. Relationally encoded links and the rhetoric of hypertext. *Proceedings of the Hypertext '87 Conference*, (Chapel Hill, N.C., November), pp. 331-344.

Meyrowitz, N. 1986. Intermedia: The architecture and construction of an object-oriented hypermedia system and applications framework. *Proceedings of the OOPSLA '86 Conference*, (Portland, OR, September), pp. 186-201.

Nelson, T. H. 1965. A file structure for the complex, the changing, and the indeterminate. *Proceedings of the 20th National ACM Conference*, pp. 84-100.

Nelson, T. H. 1987. *Literary Machines*, Edition 87.1. Available from the Distributors, 702 South Michigan, South Bend, IN 46618.

Slatin, J. 1988. Toward a rhetoric for hypertext. Paper presented at the Hypermedia '88 Conference, (Houston, TX, August).

van Dam, A. 1988. Hypertext '87: Keynote Address. *Commun. ACM*, 31, 7, (July), 887-895.

Walker, J. H. 1988. The role of modularity in document authoring systems. *Proceedings of the ACM Conference on Document Processing Systems*, (Santa Fe, NM, December), pp. 117-124.

Yankelovich, N., Haan, B., Meyrowitz, N., and Drucker, S. 1988. Intermedia: The concept and the construction of a seamless information environment. *IEEE Computer*, 21, 1, (January), 81-96.

Suggested Hypertext Bibliographies

The following references are comprehensive bibliographies of the hypertext area.

Boston Computer Society. 1988. *A Bibliography for Hypertext.*

Leggett, J. J., Schnase, J. L., and Kacmar, C. J. 1992. Working bibliography of hypertext. Department of Computer Science Technical Report No. TAMU 92-001, Texas A&M University, College Station, Texas.

Yankelovich, N., and Kahn, Paul. 1989. Hypermedia bibliography. Available from the Institute for Research and Information and Scholarship (IRIS), Box 1946, Brown University, Providence, RI, 02912.