Encyclopedia of Computer Science

FOURTH EDITION

Editors:

Anthony Ralston
Edwin D. Reilly
David Hemmendinger

nature publishing group

Websites

A comprehensive directory of computer-based activities related to English literature may be found on the World Wide Web at http://gwis2.circ.gwu.edu/~scottlib/english.html.

An account of one scholar's utilization of the computer in literary analysis and his reflections on the implications of that activity can be found at http://ilex.cc.kcl.ac.uk/wlm/essays/What/.

Joseph Raben

HYPERMEDIA

See MULTIMEDIA.

HYPERTEXT

For articles on related subjects *see* BUSH, VANNEVAR; DIGITAL LIBRARIES; HUMAN FACTORS IN COMPUTING; INFORMATION RETRIEVAL; MARKUP LANGUAGES; MULTIMEDIA; USER INTERFACE; and WORLD WIDE WEB.

Hypertext is both the concept of interrelating information elements (linking pieces of information) and the name used to describe a collection or *web* of interrelated or linked nodes. (An information element or *node* can range from a single idea or *chunk* to an entire document.) A hypertext system allows an author to create the nodes and the links among them, and allows a reader to *traverse* these links, i.e. to navigate from one node to another using these links. Typically, hypertext systems mark link access points or *link anchors* in some manner within a node when displaying it on a computer screen (e.g. underlined—often blue—text displayed within documents on World Wide Web browsers). When the user selects the link marker, e.g. by clicking on it with a mouse, the hypertext system traverses to and displays the node at the other end of the link. If a single link marker represents multiple links, the hypertext system may present the user with a list of available links. (System designers may have to rank, filter, or layer this list if the number of possible links might overwhelm the reader.) Hypertext user interface design principles recommend that authors label the link marker if the link's purpose or destination is not clear. Hypertext systems include many *navigation, annotation* and *structural features* which take advantage of the node and link structure to support authors and readers. These components and features are described further below.

Many people consider the terms *hypertext* and *hypermedia* synonymous. Nominally hypertext refers to relating textual elements, while hypermedia encompasses relationships among elements of any media type. The underlying concept is identical, though nontextual hypermedia is generally more difficult to implement.

Hypertext, when well designed, enables people to read, comprehend, and write information more effectively than traditional documents. People typically read documents from start to end, i.e. in a *linear*, sequential manner. Paper constrains (and encourages) authors to present information in a linear format. By tradition, as well as from the need to print, many write computerized documents in a linear format. Hypertext frees readers and authors from this constraint. Authors can structure information as a web of information chunks and interrelating links. For example, authors could place their main idea or an overview in an entry-point chunk (node) with multiple links connecting logical next steps or related tangential information chunks. Presenting information as a web enables readers to access information in the order most appropriate to their purposes.

Hypertext links and structural features enable authors to provide a rich context of related information around elements. As a side effect, through the process of crafting an information structure of nodes and links, authors often come to understand the information better. For readers, freedom of access within a web enhanced with contextual information provides a richer environment for understanding the information they find. Hypertext also enhances comprehension because it mimics the associative networks that people use cognitively to store and retrieve information.

Hypertext concepts can supplement other computer applications. Applications can provide links within screens and documents to related information, and can implement hypermedia navigation, annotation and structural features to provide additional context and increase comprehension. Hypertext constructs can be predefined or can be generated dynamically as an application executes.

Historical Perspective

Hypertext is not a new concept. In 1945 Vannevar Bush proposed the Memex system, which would maintain links and annotations over printed materials (Bush, 1945). In the 1960s Theodor Holmes Nelson coined the terms hypertext and hypermedia. Nelson envisioned a worldwide integrated document base of hypertext-linked information that all people would be able to access and work within. He also laid out the specifications for his Xanadu hypermedia system in the 1960s (though the first working prototype appeared only in the late 1990s). Douglas Engelbart demonstrated NLS/Augment, the first distributed, shared-screen, collaborative hypertext system at the 1968 Fall Joint Computer Conference. At the same time, Andries Van Dam was developing Fress at Brown University. Other prototype and commercial hypertext systems appeared in the 1970s and 1980s, including Document Examiner, gIBIS, Guide, Hypergate, HyperTIES, Intermedia, MacWeb (by LIRMM), Max, Neptune, NoteCards, PHIDIAS, StorySpace, Writing Environment, and ZOG/KMS. (Apple Computer's HyperCard is not a true hypertext system, as it lacks many of the navigation, annotation, and structural features that characterize true hypertext systems. Several hypertext applications, however, have been built using HyperCard's infrastructure.) Intermedia was used for many years at Brown University to teach English and biology courses (Yankelovitch *et al.*, 1988). The first major hypertext research conference, ACM Hypertext '87, was held in April 1987 in Chapel Hill, USA. Biannual at first, the hypertext research community now meets every year at this conference.

Until the 1990s, the vast majority of hypertext systems were standalone nondistributed systems. In the early 1990s, HyperG (now called HyperWave) and the World Wide Web (WWW) appeared—the first distributed hypertext systems since NLS/Augment to take full advantage of the Internet. Like HyperCard, the WWW [or more precisely, the combination of the HTTP Internet protocol and HTML (*see* MARKUP LANGUAGES)] lacks many hypertext features; though again, many true distributed hypertext features and applications have been built upon its infrastructure. As the WWW gained prominence, the research-oriented hypertext systems of the 1990s increasingly migrated to or were developed directly for the WWW environment. ACM (*q.v.*) SIGLINK was formed in the early 1990s (renamed SIGWEB in 1998). SIGWEB is the major professional society for researchers and practitioners interested in hypertext and the Web. Seminal works in the hypertext field include (Nelson, 1974, 1981; Conklin, 1987; Halasz, 1988; Malcolm *et al.*, 1991).

Hypertext Components

Nodes and Composites

Hypertext components include nodes, links, link anchors, link markers, and composites. Hypertext nodes contain the content and attributes of information elements. Approximately half of the hypertext systems before the 1990s used fixed-size windows without scroll bars. Their designers believed that hypertext nodes should contain a single *chunk* of information, i.e. a single idea. The others supported entire documents as individual nodes. Many models of hypertext support *composite nodes*, i.e. a distinct set of nodes, perhaps with interrelating links. An example would be a book of separate chapters or a car of many components. The entire composite can be treated as a single high-level node. Links can be made to the composite or to any of its component nodes.

Many hypertext systems allow nodes to have *semantic types*, such as "description" or "letter of complaint," which they display with the node, as well as in maps and overviews of the hypertext web to help orient the user (*see* Fig. 1). Node types may also correspond to specific templates or forms that structure their internal content. People can use types in *structural search* to describe the nodes they seek. Several researchers have proposed taxonomies of node types. Some are detailed and domain-specific. Others are generic, used primarily for theoretical explanations, such as "proposition" and "collection."

Link Anchors and Link Markers

Links connect entire nodes, but they focus on a particular aspect of each node endpoint. *Link anchors* specify a particular target area within each node corresponding to this aspect. Users typically do not see the link anchor, which is embedded in the node's internal code (e.g. HTML) and often contains internal parameters such as how to find its link. Instead the hypertext system displays a *link marker* that users can select to activate the anchor and traverse its link. On WWW browsers a link marker is often blue underlined text, denoting a link anchor. For example, a link might connect two newspaper articles if they both mention the same person as in Fig. 1. The person's name is associated with an anchor in each of the two nodes' internal code, as shown in Fig. 2. On the computer screen, the hypertext system highlights the person's name as a link marker whenever it displays one of the articles on the screen (perhaps incorporating the link label: "another article about Ms. X"). Selecting the marker activates the anchor. The system retrieves the anchor's link, determines the destination node and displays it, scrolling to the person's name in the second article.

Many hypertext systems will highlight the *destination link marker* or target area in some special way (e.g. displaying it in a different text style or flashing a box around it several times or shading its background as in Fig. 1) to draw the user's attention to the focus of the link just traversed. Note that in this case, the system also should highlight the destination link marker as a possible *departure link marker*, so users can traverse the link in both directions. A single link anchor and marker can represent multiple links. When selected, the hypertext system displays all relevant links and the user can choose among them. A few hypertext systems support *overlapping* link anchors and markers. When the link's focus truly encompasses the entire node, then the link anchor includes the node's entire content. The hypertext system should indicate the existence of any links leading from that node, and that the entire node is the destination target area when traversing to it.

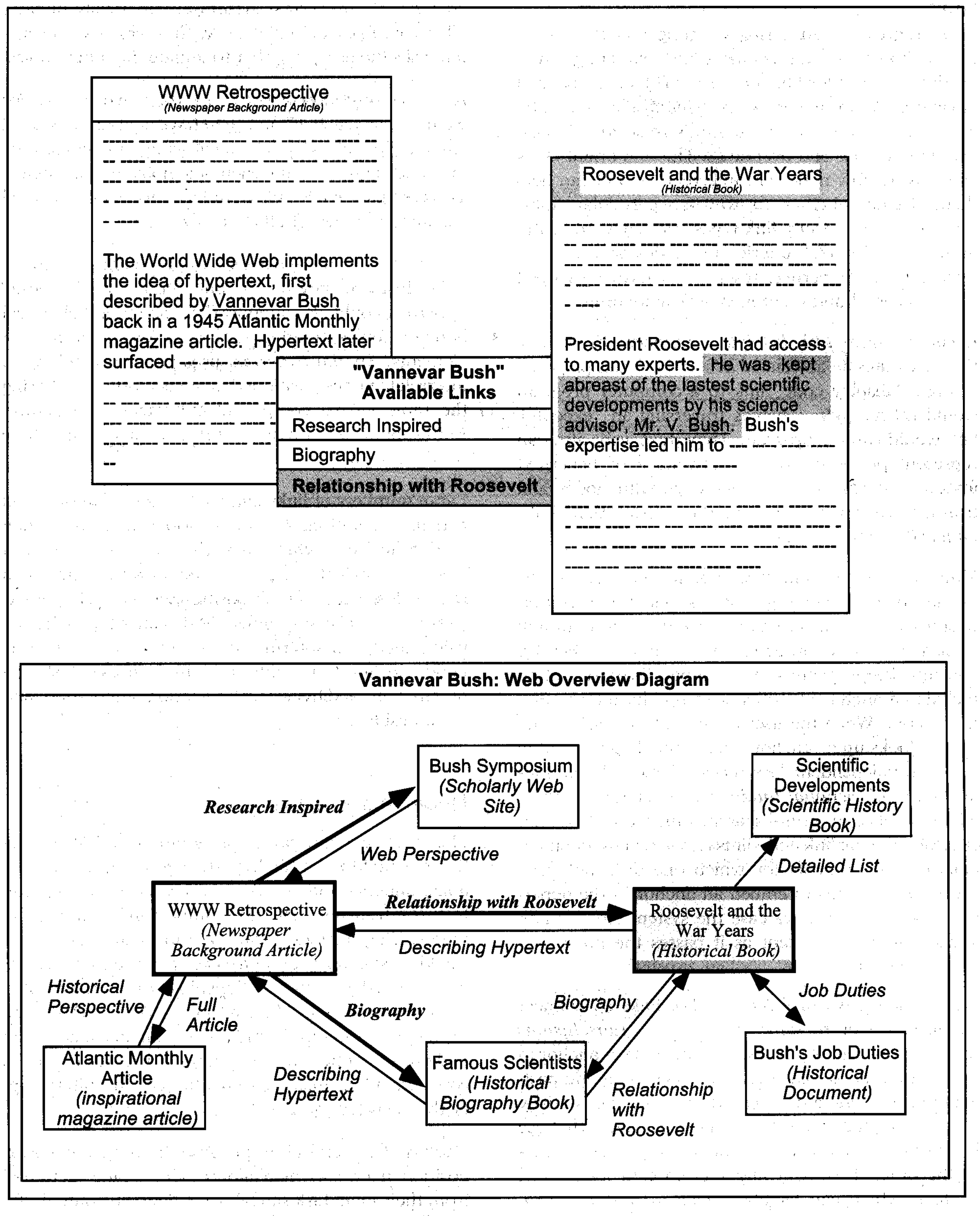


Figure 1. Traversing a link within a hypertext web.

The user has selected the link marker "Vannevar Bush" in the document node "WWW Retrospective," causing the hypertext system to display the three links for that anchor. Choosing the third, "Relationship with Roosevelt" leads to displaying the destination document node "Roosevelt in the War Years." The target area for that particular link is highlighted. (The target area contains another link marker, "Mr. V. Bush.") The bottom window contains a local overview of this Web, showing the nodes and node types within boxes, and the links and link types as arrows. The nodes and links appearing in the windows above are displayed with bold lines. The current node is highlighted in gray.

LINKS

Links represent relationships among nodes. As with nodes, links may have *semantic link types* (e.g. "supporting evidence for" or "criticism of") and associated *keywords*. Many hypertext systems display semantic link types as labels near the link's marker in nodes, and within maps and overviews. This can orient users by showing how nodes are interrelated, and can help them decide whether to traverse particular links. Users may also specify link types and keywords within *structural search* queries for nodes with specific relationships. Hypertext researchers have developed several general and theoretical link taxonomies.

Links can have *link behaviors* attached. Traversing them executes an associated action. For example, traversing a "display code" link to a computer program would list its code, whereas traversing an "execute" link would run the program. Hypertext webs also can represent processes to take advantage of hypertext features, representing process steps with nodes and transitions as links. Traversing these "transition" links *enacts* the process steps.

Many hypertext systems treat links as *first class objects* in their own right, i.e. giving links identifiers and attributes, and storing them in *external linkbases* or link databases so they can be reasoned about (*see* Fig. 2). Systems keeping links in external linkbases often include an anchor identifier with no link information in anchors. When the user selects a link marker, the system looks up its anchor's corresponding link, often using an independent *link service*. External linkbases facilitate *bidirectional linking* (traversing a link from all endpoints), structural search, *link evaluation* (e.g. checking that all link endpoints exist), and maintaining links over documents for which one does not have access permissions to embed an anchor in the departure node. In the latter case the system embeds link anchors in node content as it passes the node to a browser for display.

Transclusions, warm links, and hot links all connect two instances of the same information. *Transclusions* (or inclusions) enable the exact same node or anchor to appear in multiple places. Whereas copying and pasting creates an identical copy, transclusions essentially are pointers that connect the original to places that use it. Through transclusion, readers always have access to the original element and therefore to its original context, which can facilitate *deep intercomparison*. Xanadu's virtual document structure is built around transclusions: each document is a list of pointers to pieces of data which originate in that document or are "included" from others. *Warm links* and *hot links* are not pointers, but connect actual copies of their anchor content. With hot links, when the content of one anchor changes, the hypertext system automatically updates all other copies immediately. With warm links, the system asks the users whether to update the other copies.

An *n-ary link* interrelates more than two nodes. For example, a "teacher" node can have an n-ary link to all students in a class. A "chemical bond" link could connect nodes representing each of a molecule's elements. Traversing an n-ary link would lead to any other or all destination nodes simultaneously.

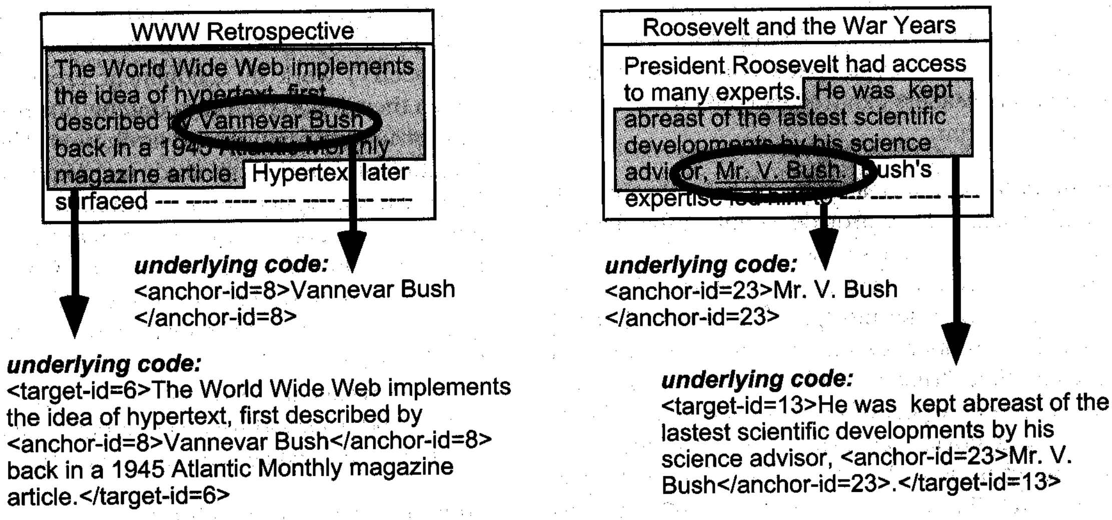
Some models of hypertext do not contain explicit links. Instead, they handle links implicitly. *Set-based hypertext* could also support the teacher and chemical bond examples, treating a class or molecule as a set of elements. *Spatial hypertext* uses pattern matching to interrelate nodes positioned near each other or sharing the same virtual composite and attribute structure. Both support other hypertext functionality using these *implicit links*.

Large numbers of links can lead to cognitive overload and disorientation. *Cognitive overload* occurs when readers feel overwhelmed with too many choices of where to navigate next. *Disorientation* occurs when readers lose track of their position within the hypertext web while traversing links. Both can be avoided by using good user interface design principles; semantic node and link types; filtering based on user task and preferences; and hypertext navigation, annotation, and structural features.

Hypertext Features

The hypertext navigation, annotation and structural features build upon the hypertext constructs of nodes, links, anchors and composites. Hypertext *navigation* features transport the user among information elements. They include *browsing* (link traversal), backtracking, standard *content-based query*, and structural query based on interrelationships. An example of a *structural query* is "find all nodes of semantic type 'product information' within a two-link distance of any node of semantic type 'legal advice' where one of the connecting links has the semantic type 'prohibited by' or the keyword 'urgent.'"

Backtracking serves four purposes: to return to a prior position in a web (which allows users to take "detours" from their main task safely), to review the content of a previously visited node, to recover from a link chosen in error, and as part of undoing. Backtracking differs from *undoing*, however, in that backtracking returns the reader to a previously visited node in its current state. *Parametric backtracking*, written as "go-back(X)", allows the user to specify a value for a



Linkbase					
Link ID	Link-Type	Source Node URI	Source Anchor ID	Destination Node URI	Destination Target ID
:	:	:	:	:	:
12	Research Inspired	http://...	8	http://...	16
13	Web Perspective	http://...	18	http://...	6
14	Relationship with Roosevelt	http://...	8	http://...	13
15	Describing Hypertext	http://...	23	http://...	6
16	Biography	http://...	8	http://...	19
17	Describing Hypertext	http://...	12	http://...	6
:	:	:	:	:	:

Figure 2. Source link anchors, destination target areas and the linkbase.

This shows the underlying link anchors and destination target areas from Figure 1, and their corresponding entries in the linkbase. The linkbase contains the three links for anchor #8, which was selected in Figure 1, from which the user selected link #14 leading to destination target area #13 in document node "Roosevelt and the War Years." Note that destination target area #13 contains source anchor #23. Selecting anchor #23 would traverse link #15 back to target area #6 in the original document node "WWW Retrospective." Destination target area #6 contains source anchor #8.

node attribute in parameter *X*, which causes the system to backtrack to the most recently departed node with that attribute value. *Conditional backtracking*, written as "go-back(query-expression)" evaluates an arbitrary *query-expression* and returns the reader to the most recently departed node satisfying it. Displaying a *history list* of previously visited nodes, and then selecting one of these, *backjumps* the user to that node. When implementing the history list as a stack (*q.v.*), backjumping removes all the in-between nodes from the history list. When implementing the history list as a *session log*, backjumping adds the prior node to the list (though some systems then remove it from its prior position in the log). Authors can construct guided tours and trails directly from session logs.

Annotation features include bookmarks, landmarks and comments. *Bookmarks* are one-way links from the reader's desktop to nodes they wish to access easily. *Landmarks* are one-way links from everywhere to a specific place, such as a home page. Landmarks typically remain permanently in view. Authors create landmarks, whereas readers create bookmarks. Bookmarks are essentially personal landmarks.

Structural features enable navigation through local and global overviews, and along recommended paths and guided tours of interrelated items. *Overview diagrams* (or maps) provide a graphical view of the hypertext web, usually with the nodes as icons and the links as arrows, often showing node names, and node and link semantic types. *Global overview* diagrams provide an overall picture; *local overviews*, such as Fig. 1, provide a fine-grained picture of a node's local neighborhood. Both provide spatial context and reduce disorientation. Readers can traverse directly to a node by selecting its icon.

Trails or *paths* connect a chain of links through an information space. They provide a context for viewing and understanding a series of nodes. They can record a path of information to remember or share. They can suggest a subset or ordering of nodes within a hypertext web, which can reduce cognitive overhead. Authors can prepare multiple "recommended" trails, each focusing on a different aspect of a web or tailored to different readers (a novice, an expert, etc.) *Guided tours* are restricted trails with link anchors that lead away from the trail dimmed or hidden. Users have to suspend or exit the tour to access these. Trails can contain *branches* allowing the reader to choose among subpaths. Hypertext systems often display and highlight trails in an overview diagram, so readers can maintain their orientation.

In multi-user hypertext systems, each hypertext construct and feature can have *access permissions* specified, e.g., for being created, modified, deleted, linked to, and commented upon by an individual, work group, or the general public.

The hypertext community strongly believes in *the reader as author*. All readers should be able to add their own annotation and structural features, as well as additional links to all nodes in any hypertext web.

Degradation or loss of relevance poses an important problem for hypertext links, comments and other features. No mechanism exists for determining when a comment, for example, has lost its relevance or correctness.

Hypertext Subfields

Hypertext research comprises several subfields: adaptive hypertext; hypertext design, evaluation, writing; hypertext functionality; open hypertext systems; and hypertext standards. *Adaptive hypertext systems* employ a user model to customize node content and filter the available link set. Adaptive hypertext systems try to guide users towards interesting and relevant information and shield them from irrelevant information. *Hypertext design* concerns analysis and design methodologies for creating hypertext systems. Hypertext design differs from standard design techniques due to its emphasis on links as first class objects and navigation. *Evaluation* techniques judge the ability of users to navigate effectively within a hypertext web and remain oriented when jumping into the web at random (e.g. to a node found by a search engine). Authors of *hypertext literature* (novels, short stories, and poetry) work in a nonlinear creativity space in which they design not only content, but also link structure, structural features and navigation. The Hypertext Functionality workshops study techniques for applying hypertext constructs and features to the everyday, nonhypertext applications found in business, engineering, and personal applications. The Open Hypertext Systems group studies ways for different hypertext systems to coordinate information and services over the Internet. (Such interoperability, for example, would allow one user to start a link using one hypertext system and a second user to complete the link using a different hypertext system on a separate computer.)

Four major *standards* efforts exist in the hypertext community. HyTime is an extension of SGML for specifying common hypertext concepts that sets of multimedia SGML documents can share. Due to its complexity, HyTime has never been widely used for hypertext. The *Dexter* reference model provides a detailed model of hypertext components. Several hypertext systems are based on Dexter. The Open Hypertext Systems group is developing an Open Hypertext Protocol for passing messages and sharing hypertext services among interoperating hypertext systems. Several hypertext systems have been made OHP-compliant. Many hypertext researchers are active in the World Wide Web Consortium's various standards efforts. The WWW's XML, XLink, and XPointer standards are being designed to support more sophisticated hypertext constructs and features on the World Wide Web.

Hypertext fills a conceptual niche related to, but separate from, fields such as databases, digital libraries, document management, information retrieval, multimedia, object-orientation, semantic networks, user interface design, and the World Wide Web. Many of these either use hypertext, complement hypertext or can be used to implement hypertext components and features.

Bibliography

1945. Bush, V. "As We May Think," *Atlantic Monthly*, **176**, 101–108. Also at `http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm`.

1974. Nelson, T. *Computer Lib/Dream Machines*. Chicago, IL: Hugo's Books. Revised edition (1987). Redmond, WA: Microsoft Press.

1981. Nelson, T. *Literary Machines*. Swarthmore, PA (privately published). Revised (1993, Version 92.1). Sausalito, CA: Mindful Press.

1987. Conklin, E. J. "Hypertext: A Survey and Introduction," *IEEE Computer*, **20**, 9, 17–41.

1988. Halasz, F. G. "Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems," *Comm. of the ACM*, **31**, 7 (July), 836–855.

1988. Yankelovich, N., Haan, B. J., Meyrowitz, N. K., and Drucker, S. M. "Intermedia: The Concept and the Construction of a Seamless Information Environment," *IEEE Computer*, **21**, *1* (January), 81–96.
1991 Malcolm, K. C., Poltrock, S. E., and Schuler, D. "Industrial Strength Hypermedia: Requirements for a Large Engineering Enterprise," *Hypertext '91 Proceedings*, 13–24. New York: ACM Press.
1997. Snyder, I. *Hypertext: the Electronic Labyrinth*. New York: New York University Press.

Websites

Hypertext Functionality. `http://www.cs.nott.ac.uk/~hla/HTF/`.
Open Hypertext Systems. `http://www.csdl.tamu.edu/ohs/`.
Open Hyptertext Protocols. `http://www.w3.org`.
Sigweb. `http://www.acm.org/sigweb/`.

Michael Bieber