

# Some Hypermedia Ideas for the WWW

Michael Bieber

Institute for Integrated Systems Research;  
NJ Center for Multimedia Research; and  
Nat'l Center for Transportation & Industrial Productivity  
New Jersey Institute of Technology  
Newark, New Jersey 07102, USA  
bieber@cis.njit.edu - <http://megahertz.njit.edu/~bieber>

Fabio Vitali

Dept. of Computer Science  
University of Bologna  
Piazza di Porta San Donato 5  
I-40121 Bologna BO, Italy  
[fabio@dm.unibo.it](mailto:fabio@dm.unibo.it)

Helen Ashman

Information Technology Division  
Defence Science and  
Technology Organisation  
P.O. Box 1500  
Salisbury, S.A. 5108, Australia  
[helen.ashman@dsto.defence.gov.au](mailto:helen.ashman@dsto.defence.gov.au)

V. Balasubramanian

E-Papyrus, Inc.; and  
Graduate School of Management  
Rutgers University  
Newark, NJ 07102, USA  
[bala@pegasus.rutgers.edu](mailto:bala@pegasus.rutgers.edu)  
<http://eies.njit.edu/~333/bala.html>

Harri Oinas-Kukkonen

Dept of Info. Processing Science,  
University of Oulu  
Linnanmaa  
FIN-90570 Oulu, Finland  
[hok@rieska.oulu.fi](mailto:hok@rieska.oulu.fi)  
<http://rieska.oulu.fi/~hok/>

## Abstract

*World Wide Web (WWW) authors must cope in a hypermedia environment analogous to second-generation computing languages, building and managing all hypermedia links using simple anchors and single-step navigation. We present a set of third- and fourth-generation hypermedia functionalities, which WWW developers should consider. We ground our discussion in the hypermedia research literature, and illustrate both from existing implementations and a running scenario. We also give some direction for implementing these on the WWW.*

## 1. Introduction

The World Wide Web's (Berners-Lee, Cailliau, Luotonen, Nielsen and Secret, 1994) brilliant success derives in part from global scale and simple access to information. Yet, Web authors must build and manage (structure and present) all hypermedia links using simple anchors and single-step navigation. In some sense, they must cope in a hypermedia environment analogous to second-generation computing languages (i.e., assembler language) in that they only have relatively low-level functionalities available. Following this analogy, we believe Web authors and readers alike would benefit from the equivalent of third-generation and fourth-generation hypermedia authoring environments. Third-generation languages, such as C and Pascal provide high-level programming concepts and data structures that

enable programmers to declare procedural steps without having to worry about low-level implementation details. Fourth-generation packages such as spreadsheets, word processors and report writers let the programmer (often an end-user) concentrate on *what* they want to achieve instead of the process of *how* the underlying program actually produces it. In this paper we give a vision of third- and fourth-generation hypermedia by describing several high-level hypermedia constructs which we believe the Web could provide today. We describe each, give examples from research or commercial systems (on and off the Web), and note implementation considerations. Third-generation features allow authors to specify hypermedia features at a higher level of abstraction than the Web's current untyped, single-step links and nodes. Fourth-generation refers to high-level features and fourth-generation style authoring environments, in both of which authors (or readers) specify "what" they want instead of the detailed steps of "how" to implement it. We hope this paper will encourage builders of Web technology to incorporate high-level hypermedia features, so everyday authors and readers can utilize them. In §2 we describe the concept of hypermedia. In §3 we introduce our scenario and present our set of hypermedia functionalities. In §4 we note some additional hypermedia features, and conclude.

A word on our terminology from the hypermedia research literature. First the term *hypermedia*, which nominally applies *hypertext* concepts to multiple media. Hypermedia researchers view the terms as synonymous and use them interchangeably. We separate users into

nominally applies *hypertext* concepts to multiple media. Hypermedia researchers view the terms as synonymous and use them interchangeably. We separate users into two categories, *readers* and *authors*, in order to highlight how each interacts differently within a hypermedia system. We distinguish authors from the *system developers* who develop browsing software and other hypermedia environments. Readers *traverse* links during the act of browsing. When appropriate, we distinguish the underlying *link anchor* from the manifested *link marker* displayed on the screen. Link markers visibly indicate a link's presence. Link anchors contain parameters and other internal information, which users do not see. *Nodes* represent the documents or primary content containers (Halasz and Schwartz 1994) in a hypermedia system.

## 2. Hypermedia

While many people think of hypermedia in terms of the World Wide Web, hypermedia research has been ongoing since the early 1960s when Engelbart developed NLS, a multi-user, distributed hypertext system (Engelbart and English 1968). Since then, the hypermedia research community has been developing a wealth of features, systems, guidelines, frameworks and theory focusing on structuring, presenting and accessing interrelated information.

Hypermedia enables people. Hypermedia, as a concept, encourages authors to structure information as an associative network of nodes and interrelating links. This frees authors from the linear, sequential structure that dominates most printed documents. Presenting information as an associative network enables readers to access information in the order most appropriate to their purposes, freeing them from obeying the implied linear ordering within printed documents. Furthermore, many hypermedia implementations allow readers to become authors (temporarily, at least) by adding comments (annotations) upon and additional links among what they read. In all these ways the concept of hypermedia promotes options and choice.

In a larger sense, hypermedia increases comprehension (Thüring, Hannemann and Haake 1995). Through the process of structuring information as an associative network, authors often come to understand that information better. Comprehension also increases through the enriched context that comes from sophisticated navigation support and supplemental relationships. For example, hypermedia encourages authors to provide multiple relationships around a piece of information, which readers can access directly. Thus,

for readers, freedom of access within an associative structure enhanced with contextual support provides a rich environment for understanding the information they find.

In his seminal hypermedia survey, Conklin identifies two major dangers of free-formed hypermedia access within an associative network: *disorientation* and *cognitive overhead* (Conklin 1987). Conklin defines disorientation as "the tendency to lose one's sense of location and direction in a nonlinear document," using the expression "lost in space" to describe it. He defines cognitive overhead as "the additional effort and concentration necessary to maintain several tasks or trails at one time." Cognitive overhead refers to the reader's ability to follow links related indirectly to the current reading task (on a purposeful tangent or detour, or by accident), as well as the need to follow several interconnected paths to visit as much of the associative network as necessary. Furthermore, cognitive theory reminds us of the overhead that comes from needing to choose among multiple links, especially for novices not familiar enough with a domain to decide among these easily (Wurman 1989). While many hypermedia researchers believe that disorientation and cognitive overhead are problems inherent in hypermedia, proper implementations of advanced hypermedia features will alleviate these problems as well as provide readers with a richer information environment.

In this paper we propose that incorporating a range of high-level hypermedia features will enhance Web applications even further. Thus, while we do discuss many of the advanced hypermedia features in terms of how they reduce cognitive overhead and disorientation, we also fully motivate each by showing many other ways that they enhance applications, both on the WWW and in other information systems.

## 3. High-Level Hypermedia Features

Consider the following scenario:

A technical documentation shop obtains product specifications from product designers and produces four types of documentation: printed manuals bundled with the product; on-line manuals for World Wide Web access; official documentation (such as blueprints, test results and legal documents for patents and government approvals); and internal documentation for new developers, help desk employees, and exploiters of undocumented features. Furthermore, if closely related versions of the same products exist (for instance, software versions for different platforms, or concerning replacement parts for similar mechanical objects), the

documentation effort should not be duplicated; differences among the various versions should not prevent drafting identical parts of the documentation in common.

The amount of information delivered to each of three audiences varies based on access privileges. For example, information to the general public may include description of products, their suitability for people in various circumstances, introductory manuals for use and reference. Help desk consultants will have access to more detailed technical documentation, so that they can answer tricky questions. Internal developers will have access to even more detailed documentation, such as product specifications and production code. The system also will provide tailored views of the documentation for different users in these three audiences.

Information can take the form of on-line brochures, instruction kits, application forms and short synopses, and will include interrelationships (hypertext links). All product marketing information has to be approved by appropriate legal authorities within the company. Versions of the material presented on-line have to be preserved and remain accessible for regulatory reasons.

We shall refer to this technical documentation shop scenario throughout §3. We begin in §3.1 with issues concerning link and node types. §3.2 covers issues concerning distributed collaborative hypertext. §3.3 describes some hypermedia tools assisting users in orientation and navigation.

### 3.1. Typed links and nodes

Typed nodes and links help authors organize information more effectively, and lend context for readers. Typed link labels such as "explanation," "further details," "contrasting argument," etc., convey the relationship between the link's destination and the current node. Similarly, typed node labels categorize a node's contents.

In our technical documentation shop example, semantic link labels could distinguish all information related to a product: "description," "diagrams," "component details," "medical warnings," "legal information," etc. Readers normally would access these from an index or table of contents, but the documentation shop can embed these hypertext links directly to the exact section, paragraph, line or even word to which the information pertains. Users presumably will only be interested in a subset of this information, and thus will find the semantic type important in determining whether to traverse a given link. Semantic links can also connect the different levels

of detail within the documentation base for the different users: "full details," "public release," "internal specifications," "known problems," "troubleshooting information," etc.

The Web has always had a mechanism for specifying link types, and even has a set of types, which slowly is becoming an accepted standard (Maloney and Quin 1996). Few browsers, however, take these into account to help guide navigation. Context information aside, the vast majority of WWW authors rely on the content of a link marker to convey 100% of the link's purpose and destination. Thus we experience many Web users complaining that sometimes they do not really know where a link will take them. This adds to cognitive overhead for the reader. We view this, in part, as both a user interface design question and a hypermedia design question.

Thüring et al. (1995) list providing semantic link labels as their *first* principle of hypermedia design. But the hypermedia research literature has produced many link taxonomies. Textnet (Trigg et al. 1986), SEPIA and MUCH (Wang and Rada 1995) enforce a limited set of link types. Other systems, such as LIRMM's MacWeb and NoteCards permit arbitrary link labeling. Arbitrary labeling gives authors more flexibility to express exact meaning, but carries the danger of inconsistent type names (Rada 1990).

Other link taxonomies are designed purely for the benefit of researchers and system developers, which systems will never display to readers (De Rose 1989) (Parunak 1991) (Rao and Turoff 1990). These include categories such as extensional, inclusive, intensional, implicit and isomorphic (De Rose 1989). System developers can use these taxonomies in determining appropriate navigation strategies and ways of conveying context for different categories of links underlying sophisticated hypermedia environments. Rao et al. (1990) also include a theoretical node taxonomy for the same purpose.

Wang and Rada point out that authors often do not take the time to incorporate link types, even when appropriate (Wang et al. 1995), so system designers providing semantic typing should make them as easy to specify as possible. Even when the reader's hypermedia environment does not display node and link types, Nanard et al. encourage system designers to facilitate typing as far along the design process as possible, to assist authors in structuring and organizing their work (Nanard and Nanard 1995).

Link types can be shown as labels in an overview diagram, as in SEPIA (Streitz et al. 1992) (Thüring et al.

1995), LIRMM's MacWeb<sup>1</sup> (Nanard and Nanard 1991) (Nanard and Nanard 1993) (Nanard et al. 1995), and gIBIS (Conklin and Begeman 1989). Link labels could also pop up when the user moves the mouse cursor over them, as in the hypermedia version (SIGLINK 1994) of papers from the August 1995 issue of the *Communications of the ACM* (Bieber and Isakowitz 1995).

The *Communications of the ACM* system uses World Wide Web technologies to implement these functionalities. The authors have used low level functionalities (frames and the Javascript programming language). A third-generation type of authoring tool, on the other hand, would have given authors data structures providing for semantic types, and the browsing tool would have displayed node and link labels automatically in overview diagrams and in other content displays. A fourth-generation authoring environment would add a magnitude of functionality. Authors would be able to choose from a set of established node and link types, and would have a high-level interface for creating new ones (or new sets for a specific group of other authors). The system would associate semantic types automatically with appropriate *browsing semantics* (Garzotto, Mainetti and Paolini 1996) (Schwabe, Rossi and Barbosa 1996), i.e., the behavior associated with objects of that semantic type when users navigate over them.

**3.1.1. Annotation.** Hypermedia researchers have always considered private annotations (comments) a basic right for hypermedia readers as well as a basic tool for collaboration and exchange of ideas. Almost every major hypermedia system provides support for annotations: KMS (Akscyn, McCracken and Yoder 1988) (Yoder, Akscyn and McCracken 1989), Intermedia (Catlin et al. 1989), Aquanet (Marshall, Halasz, Rogers and Janssen 1991), the Virtual Notebook System (Burger, Meyer, Jung and Long 1991), StorySpace (Joyce 1991), etc.

While WWW committees have discussed annotations (La Liberte 1994) (Gramlich 1994), few browsers support annotation, as it is difficult to embed annotation links within HTML documents belonging to others. Mosaic 2.4 (National Centre for Supercomputing Applications 1995) provides an annotation facility. Unfortunately however, due to implementation restrictions, readers cannot associate an annotation with the specific spot in the document upon which the reader

wishes to comment. Schloss (1996) presents an annotation mechanism on the World Wide Web for augmenting electronic commerce applications with advisory systems. His advisory service enables third parties to provide supplementary information to support, augment or critique the contents of Web pages.

**3.1.2. Transclusions, warm links and hot links.** Instead of linking one node to another, some link types enable referred-to data to appear in a host document. Transclusion, warm links and hot links all connect two or more occurrences of the same information, allowing for the same information to appear in different contexts without replication or with controlled redundancy.

Nelson (Nelson 1987, Nelson 1995) proposed *transclusions* (or *inclusions*) as a mechanism for the same chunk of information (document content) to appear in multiple places. Whereas copying and pasting creates an identical copy, a transclusion acts similar to a pointer that connects to the original copy in all places that use it. Transcluded data is *alive*, still connected to the original and automatically updated. Through transclusion readers always have access to the original and therefore to its original *context* (through a context link). The virtual document structure of Xanadu, Nelson's system, 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.

In the context of our technical documentation shop, transclusions create a reliable and maintainable set of similar documents (for instance, accounting software documentation customized for clients in a particular state in the United States). Each document contains the same base information, together with different parameter settings for each user. The settings indicate the appropriate portions to transclude when presenting the document. Storing each document as a separate, complete entity would mean redundancy and complex update whenever a fairly common clause (e.g., a feature) in many needs to be changed. Widespread transclusion could reduce or eliminate redundant storage and greatly simplify the update process. Authors would specify documents in terms of transcluded elements. For example, a document may comprise standard clauses 1 to 12, clause 13 from document Y, and a special clause 14 belonging to only this kind of document. The same thing can be accomplished by component documents and republishing all document that use a component when that component changes. (Xanadu's implementation of transclusion inherently incorporates versioning (Vitali and Durand 1995), which would satisfy the legal requirement to maintain some old document versions.)

---

<sup>1</sup> MacWeb is a knowledge-based hypertext system under development at the Laboratoire d'Informatique, de Robotique et de Micro-Electronique de Montpellier, France (LIRMM) since 1989. It has no relationship to the World Wide Web client developed later with the same name.

*Warm and hot links* are relationships that create a channel between the two end-points, through which data flows from one document to the other (Meyrowitz 1989). Warm and hot links are not pointers, but actual copies of the data, which can update themselves, either after a user's request (warm links) or automatically when a change is detected (hot links). Curiously, while few existing hypermedia systems have ever implemented this kind of links (Catlin et al. 1989), implementations have come from an important class of commercial systems: operating systems. Publish/subscribe in the Macintosh operating system and dynamic data exchange (DDE) in Microsoft Windows allow applications to create live channels for displaying and updating portions of data from one application to another.

Transclusions and warm linking differ basically only on implementation details. Transclusions are pointers to data, while with warm and hot links the host document contains both a copy of the data and a reference to the original source. Simple references avoid duplications and redundancies, while copies allow improve reliability and availability—the system could still display a (possibly outdated) copy even if the original document were deleted or inaccessible).

### 3.2. Distribution issues

While the WWW was born as (and to a large extent, still is) a distributed delivery medium, collaboration, on-line authoring and other new application environments will increase the need for sophisticated multi-user solutions that can scale to the size of the Web. Here we describe a few hypertext-related issues addressing distribution and multiple users.

**3.2.1. External link database.** Embedding link endpoints in HTML presupposes that the link creator has write access to the information being linked. External links stored separate from the document content in external *linkbases* (Davis 1995a) (Davis 1995b) can be used to provide hypermedia where write access is not available, or when user tailorability of links is supported. A typical situation would be where users want to apply their own link specifications to data from a remote site or from read-only media, or where documents being linked must remain inviolate. For example, the technical documentation shop may wish to link requests, orders and other correspondence to the clients' files. For validity, these external communications should not be altered in any way, hence links coming from them must be maintained externally. Also, for legal purposes, some documents must remain unchanged. For example, links

to previous versions of a documentation or other product information must be maintained "as is" in case of future lawsuits concerning the literature. Linkbases are maintained external and independent of all the documents their links connect. This permits users of the Microcosm system, for example, to maintain their own private links in their own linkbases, which they carry with them on floppy disks (Davis, Hall, Heath, Hill and Wilkins 1992). Their private linkbase is read with all the other accessible public linkbases to determine the links available on a document.

The field of *open hypertext systems* uses external links for linking to data owned by legacy applications and other non-hypermedia applications without disturbing the original format of that data (Davis 1995a) (Davis 1995b) (Kacmar 1996). External link databases are being trialled on a larger scale in the Distributed Link Service (Carr, De Roure, Hall and Hill 1995), in which databases of precomputed links are published along with documents.

Independence between the document and links carries the danger that updating information can desynchronize the link's reference in a linkbase and its new location in its document. Byte-offsets are the most fragile of these location systems, since offsets fail after even minor modifications to the content (for instance, offsets are invalidated after any insertion or deletion preceding them). Before using it, one needs to check whether the link is still consistent, i.e., whether the nodes still exist, whether they are still reachable, and whether the stored references still point to the correct location within the nodes. Furthermore, one might wish to check whether the link still has a reason to exist, i.e., whether the changes in the documents have made the link useless, inappropriate or wrong. These problems arise when there is loose control between the link and the connected nodes, i.e., when a change in one of them does not necessarily imply a modification in the others. Named locations (as in HTML named anchors) are extremely robust to modifications, but still require the authors of the link to have write access to the destination document, and are thus not applicable in this context. Dynamically computed links are also robust, because, even if the position of the end-point may change, a string-matching algorithm can easily retrieve the new position. Vanzyl, Cesnik, Heath and Davis (1994) proposed heuristic techniques to retrieve the current position through pattern matching. While simple and powerful, pattern matching is not completely reliable, since it fails in many common situations (for instance, to retrieve any but the first of multiple occurrences of the same text).

Versioning systems (Maioli, Sola and Vitali 1993) (Nelson 1987), on the other hand, are able to check what

changes have been performed on a document making it possible to compute current offsets based on old ones. In fact, the Xanadu addressing schema, heavily based on versioned transclusions, purportedly never runs into the link update problem because original offsets are always valid regardless of how a document is modified (Nelson 1987). If properly implemented, therefore, a versioning system may become the most reliable system for external links. Authors, however, must then use only versioning-aware tools when creating and modifying documents. A versioning mark up language for the WWW has been proposed (Vitali et al. 1995), and its implementation in HTML editors and browsers would surely contribute to a natural and widespread use of reliable update techniques.

**3.2.2 Hypermedia access permissions.** Creating views and only permitting certain readers to access certain information requires both customization based on user role and access permissions. Besides standard read and write access permissions on documents, system administrators may consider analogous read and write permissions for links and annotations. For certain documents one could allow link authoring and update while prohibiting write access to the underlying content. One also could restrict author access to certain node and link types. Furthermore, one could regulate link and annotations to specific views.

With personal and workgroup annotations, annotation authors implicitly are trusted once they log onto the supporting network, hence there is little need for annotations to be vetted for suitability. However a public annotation facility, however, may or may not permit annotations to be added by an unvetted public. Thus systems need to distinguish read-only public annotations from unrestricted public annotations. For example, a site may publish a database of annotations or links which cannot be augmented by the public, or it may permit the public to add publicly-readable links and annotations to any document. These two classifications respectively are directly analogous to the concepts of moderated and unmoderated newsgroups (except that newsgroups generally include only untyped content). For this very reason, implementors of public link and annotation databases may find it profitable to consider the techniques of network bulletin board management (such as the use of "kill" files to filter out irrelevant or obnoxious annotations).

Within the ABC collaborative writing environment access permissions regulate group collaboration (Schackelford et al. 1993). Turoff has developed twenty-five kinds of access permissions (as well as "tickets" to override them) for computer-mediated communications, many of which would apply to

collaborative hypermedia applications (Turoff 1991). Mosaic's annotation facility supports three classifications: personal, workgroup and public annotations, and later versions support workgroup annotations through a server. The Distributed Link Service implemented a similar approach (Carr et al. 1995). In each case, the document server and link server can be completely different entities. This requires the link server to employ external link databases.

**3.2.3. Computed, personalized links.** A powerful feature of high-level hypermedia is the automated creation of private links from computations. Readers specify their own computations in order to create the private links, that is, the reader can tailor their own link computation specifications. This allows readers to create entire collections of links which are specialized to their needs.

For example, a designer in the advertising or graphics departments may wish to create a personalized link computation specification connecting every portion of any document that he or she designed (perhaps in a "best to worst" trail (see §3.3.2). Alternatively the technical documentation department may wish to link every document sent to a client. Instead of creating this type of link by hand every time a document is completed, the department sets up a sort of "standing order" with the hypermedia system, so that it will create these links automatically.

The key issue in computing private links is that the hypermedia system treats the computation specifications as external data. Hence they can be manipulated without the need to recompile (and hence alter) the hypermedia system software. This allows both readers and system administrators to maintain link computation specifications most relevant to their needs. This is a form of externalized link database, except that the stored objects are not links but *specifications* for computing private links.

Tailoring link computation specifications is not well-supported in general. Few hypermedia systems permit users to create their own link computation specifications or permit administrators to create link computation specifications for corporate use. Where this has been enabled, specifications are expressed in some formal language. One implementation of tailorable link types uses a relational database to store these specifications (Verbyla et al. 1994), while the Web's tailorable link computation specifications (i.e., CGI scripts and mobile code applets) are usually stored in directories on the host machine operating system. Alternatively, computation specifications can be attached to classes of objects, and while the reader may not necessarily be able to alter the

class (i.e. the domain to which the link type applies), it is possible to edit specifications for computing link destinations (Monnard and Pasquier-Boltuck 1992) (Rizk and Sauter 1992).

### 3.3. Hypermedia Features for Orientation and Navigation

**3.3.1. Local and global overviews.** Overview diagrams (or maps) provide a view of the hypermedia node and link network from above. They show its structure in a graphical manner (e.g., with icons and arrows). Global overview diagrams provide an overall picture and can include anchors (entry points) down to local overview diagrams or single nodes (documents). Local overview diagrams provide a fine-grained picture of the local neighborhood of a node. They improve spatial context and reduce disorientation in a hypermedia network. Local overview diagrams also help minimize cognitive overhead by showing a small, relevant part of the network.

Intermedia provides overview diagrams, called Web Views, which provide both spatial context ("Where can I go from here?") and temporal context ("How did I get here?") (Utting et al. 1989). Intermedia web views are dynamically updated, local tracking maps that display all documents linked to the current one (Utting et al. 1989). In Thoth-II, the "Spiders" directed graph browser (Collier 1987) dynamically creates new objects (nodes and links) through the overview. The reader interacts with the structure being viewed, expanding a visited node to show links to other nodes which subsequently fan out to further nodes.

Overview diagrams can be generated either by reading the links between documents from a link database (Andrews, 1996), or by examining the documents themselves for embedded links. However, as soon as overview diagrams for large systems become complex they might introduce navigational problems of their own (Conklin 1987) (Nielsen 1990). Nielsen suggests features which we believe an intelligent fourth-generation overview map would provide. For example, in order to reduce the propagation of links from the local diagrams to global diagrams, weights can be assigned to the links based on their relevance to the user. Similarly, readers might want to know which material in the current subweb applies to their task-at-hand. (The scope line in Intermedia, for example, includes a count of the documents and links in the web, as in Utting et al. 1989). The overview browser could apply user profiles to the node and link attributes, perhaps querying the user for

his or her interests, and then tailor the overview automatically.

In our technical documentation shop scenario, the amount of material to be presented highlights the need for overview diagrams. A user may wish to learn more about a particular product. Is a product part of a category of products with similar characteristics? Is that category part of another higher level group of products? Is the product itself related to other products from other groups? Global and local overviews will help, as this kind of information is hard to assimilate when readers only can function down at the actual content level, but much easier when the relationship (meta)structure is shown in a graphical manner.

The World Wide Web does not have the necessary constructs to provide local and global overviews or maps for two unrelated reasons. First, while global overview maps can be implemented as imagemaps, their automatic generation would be extremely complex and troublesome, having to deal with CGI or Java applications parsing the documents for link information, and creating the image (possibly a GIF) and the imagemap table for relevant users' feedback. Second, WWW links cannot be categorized by their roles: that is, there is no easy way to differentiate between *structural links* among parts of single concept or information unit and *associative links* among related concepts. Furthermore, many HTML authoring tools do not allow authors to visualize the nodes and relationships which they are creating, although tools such as Microsoft FrontPage and NaviPress allow authors to look at local mini-webs as they are being created.

**3.3.2. Trails and Guided Tours.** In what people now recognize as the first article on hypermedia, Bush introduced the idea of associative *trails* (Bush 1945) or *paths*. Trails connect a chain of links through information spaces. They provide a context for viewing and understanding a series of documents. Trails can record a path of information that the reader may wish to remember and share with others. In addition, authors (or interested third parties) can prepare multiple "recommended" trails through an associative network focusing on different aspects or tailored to different readers (a novice, an expert, a teacher, a student, etc.) Bieber (1992) notes that analysts might use trails to document a decision analysis for a boss or for new analysts to learn. Continuity and guidance distinguish trails from random links in documents. The trail should be clearly marked, so users will know which links keep to the trail and which constitute detours from the trail. The trail designer could filter links, making only the

ones most relevant to the current item available as *detours*.

*Guided tours* restrict users to the trail, prohibiting detours. Nodes viewed during the tour will have other links dimmed or hidden. Users have to suspend or exit the tour to access these (Garzotto et al. 1996). While trails lower cognitive overhead by recommending the next logical link to take, guided tours reduce overhead further by removing all other choices. This notwithstanding, trails and tours could contain branches allowing the reader to pick one subpath over another (Thüring et al. 1995). In addition, the system could provide an overview or map, so readers can maintain their orientation along the trail or tour.

We can imagine endless opportunities to enhance our technical documentation shop scenario with trails and guided tours. The company could establish a guided tour for the general public about each of its major products and then nested guided tours (Garzotto et al. 1996) within this about particular product details. The training department could set up tutorials for new agents and those seeking advanced training in various topics in the form of trails. Trainees could follow the recommended path while having access to details and tangential information. Help desk employees could utilize a chronological trail of all requests from a particular caller (or a branched chronological trail with a branch for each product about which the caller has inquired).

Trail implementations vary in complexity. In simpler implementations, such as with Textnet, the trail is an ordered list of nodes that the user follows, i.e., Textnet trails are viewed in a special window as a list of nodes (Trigg et al. 1986). Perseus represents trails as an ordered list of icons (Mylonas and Heath 1990). In Intermedia, a path is a list of documents readers visited earlier in a browsing session and saved automatically when closing the web. The display of a path consists of a list of items showing the document name, an icon indicating the event type (opening or activating documents) and a timestamp indicating when the event occurred (Utting et al. 1989).

More advanced implementations permit the author to annotate the trail to provide fuller information about its overall purpose and execution, as well as about each of its nodes. For example, NoteCards' guided tours feature a graphical overview and customized tabletop screens at each stop along the tour (Trigg 1988) (Marshall and Irish 1989). The tour's author can add several types of meta-information in a separate area on the tabletop for the tour and for each of the *notecards* or nodes included on the tour:

In Xerox's Scripted Documents system, Zellweger has extended the notion of trails to procedural

programmable paths with active entries called *scripts* (Zellweger 1989). This is analogous to a very sophisticated macro facility in which the author builds "path macros" (Parunak 1989) in which entries can perform arbitrary actions ranging from issuing system commands to manipulating variables via an interpreter. This leads to the set of hypermedia research that considers hypermedia as a way of representing and implementing processes. Representing the process steps as nodes and transitions as links enables developers and users to augment the process with annotations and with links to related information (Scacchi 1989). Similarly, some developers structure programs as associated networks (e.g., petri nets) controlling which links to activate based on the user's previous interactions (Stotts and Furuta 1989).

Trails (and tours) on the Web tend to contain only documents owned by the trail author. Authors need write access to documents to embed "previous" and "next" links, and to otherwise annotate and tailor the documents. System developers should find it possible, however, to provide environments for authors to build trails containing any document. During execution (or perhaps setup) the trail tool would process each document to append any tour commands, supplemental links, annotations, etc., and remove any links the author wishes to filter out for tours. The environment also should build a trail map which the author could modify. System developers may wish to take advantage of the data model and comprehensive set of navigation commands for implementing traversal within single and multi-level nested guided tours, which Garzotto et al. have developed (Garzotto et al. 1996). On the World Wide Web, New York University's Information Systems Department Information System (Department of Information Systems, 1996), which was designed using the RMM formal hypermedia design methodology (Isakowitz, Stohr and Balasubramanian 1995), generates guided tours automatically from a database of course, professor and publication information.

In a third-generation style authoring environment, authors would manually place each entry in the tour list and manually tailor the contents of each tour node. The system could generate previous step, next step, first entry and exit navigation buttons automatically, as well as generate an overview map. A fourth-generation trail and tour authoring environment would take advantage of semantic types, keywords and other attributes, access permissions, views and user information to organize and tailor the tour contents dynamically.



## 4. Discussion

A. Node issues
Annotations
Typed nodes
B. Link issues
Associative and annotative links
Computed and Hand-crafted links
Dynamically computed and precomputed links
Link attributes
Transclusions
Unidirectional and bidirectional links
C: Navigation issues
Author-created landmarks
Backtracking and Interaction history
Browsers and overviews
Content-based information retrieval mechanisms
Different views of a single hypermedia network
Link traversal mechanisms
Node participation in different views
Reader-created bookmarks
Structure-based information retrieval mechanisms
Trails and guided tours
D. Miscellaneous issues
Access to heterogeneous repositories
Automatic link propagation
Hypermedia-aware user interface controls or widgets
Links to data in any application
Mapping of application objects to hypermedia objects
Standards for content markup and linking

**Table 1. Hypermedia features addressed in the HTF II workshop.**

Despite the rush to place information on the WWW, many legacy and other applications will keep their own interfaces, and will not be moved onto the Internet or an intranet. Yet their users still would benefit from hypermedia functionality (Davis et al. 1994) (Bieber 1995). Most end-users are reluctant to abandon their current non-hypermedia oriented systems in favor of hypermedia features. Thus, the myriad of today's personal, scientific and business applications, which were not designed specifically as hypermedia-oriented, should be augmented with hypermedia features. The *hypermedia functionality* approach focuses on incorporating hypermedia features into software systems so as to provide their users with an associative way of

accessing, analyzing and organizing information (Oinas-Kukkonen 1997). In sum, the benefits of adding hypermedia functionality to information system applications are three-fold. First, it provides a contextual, navigational *access* for viewing the information. Second, many believe that hypermedia representation of knowledge is in a form relatively close to the cognitive organizational structures that people use, thus supporting human *understanding* (Conklin 1987) (Thüring et al. 1995). Third, hypermedia provides a natural means for supporting *communication* between different stakeholders, e.g., authors and readers.

The papers and presentations at the *Second International Workshop on Incorporating Hypertext Functionality into Software Systems* (Ashman et al. 1996) held in conjunction with the ACM Hypertext '96 conference (Washington D.C., March 1996) addressed a wide variety of hypermedia features in software systems. Table 1 classifies these into node, link, navigation, and miscellaneous issues<sup>2</sup>. The present list is by no means complete, but rather gives one an idea of the richness and applications of hypermedia functionality research. Unfortunately, in this paper we could describe only a small, but important selection of these.

Many software systems have some basic hypermedia features, providing simple annotating and linking capabilities. For instance, Microsoft Windows' help system has simple hypertext linking and annotation capabilities, and Microsoft Word has annotation and bookmark capabilities. For many of these systems, hypermedia is integrated so seamlessly that end-users are unaware of hypermedia's presence. Still, these systems could be empowered greatly by advanced hypermedia functionality. Much research remains in integrating hypermedia functionality into all types of applications. We hope this paper will encourage builders of Web technology to incorporate high-level hypermedia features, so everyday authors and readers can utilize them.

## Acknowledgments

This research has been supported generously by the NASA JOVE faculty fellowship program, the National Center for Transportation and Industrial Productivity at the New Jersey Institute of Technology (NJIT) under grant #990905, the New Jersey Center for Multimedia Research, NJIT under grant #991967, and by a grant from the AT&T Foundation.

<sup>2</sup> We especially wish to thank Carolyn Watters from Acadia University in Canada, a participant in the Hypertext Functionality Workshop II at the Hypertext'96 conference, for coming up with the structure of Table 3's feature list.

We wish to thank the participants of the Hypertext Functionality Workshops I and II, held in conjunction with the European Conference on Hypermedia Technologies (ECHT) 1994 and the Hypertext'96 Conference, respectively. Special thanks to Alejandra Garrido, Keith Instone and Paul Kahn for their help reviewing drafts of this paper.

## References

- AKSCYN R., MCCracken D., & YODER, E. (1988). KMS: A distributed hypermedia system for managing knowledge in organizations. *Communications of the ACM*, 31 7, 820-835.
- ANDREWS, K. (1996). Applying Hypermedia Research to the World Wide Web. Workshop on Hypermedia Research and the World Wide Web, Hypertext '96, Washington.
- ASHMAN, H.L., BALASUBRAMANIAN, V., BIEBER, M. & OINAS-KUKKONEN, H. (eds) (1996). Proceedings of the Second International Workshop on Incorporating Hypertext Functionality Into Software Systems II (HTF II). <http://space.njit.edu:5080/HTFII/Proceedings.html>
- BERNERS-LEE, T., CAILLIAU, R., LUOTONEN, A., NIELSEN, H.F. & SECRET, A. (1994). The World-Wide Web. *Communications of the ACM*, 37(8), 76-82.
- BIEBER, M. (1992). Automating hypermedia for decision support. *Hypermedia*, 4(2), 83-110.
- BIEBER, M. (1995). On Integrating Hypermedia into Decision Support and Other Information Systems. *Decision Support Systems*, 14, 251-267.
- BIEBER, M. & ISAKOWITZ, T. (eds.) (1995). Designing Hypermedia Applications. Special Issue of the *Communications of the ACM* 38(8).
- BURGER, A.M., MEYER, B.D., JUNG, C.P. & LONG, K.B. (1991). The Virtual Notebook System. Technical Briefing, Hypertext '91 Proceedings, ACM, 395-402.
- BUSH, V. (1945). As We May Think. *Atlantic Monthly*, 176, 101-108.
- CATLIN, T., BUSH, P. & YANKOLOVICH, N. (1989). InterNote: Extending a Hypermedia Framework to Support Annotative Collaboration, Hypertext '89 Proceedings, ACM Press, 365-378.
- CARR, L., DE ROURE, D., HALL, W. & HILL, G. (1995). The Distributed Link Service: A Tool for Publishers, Authors and Readers. Proceedings of the 4th International WWW Conference, [http://wwwcosm.ecs.soton.ac.uk/dls/link\\_service.html](http://wwwcosm.ecs.soton.ac.uk/dls/link_service.html)
- CONKLIN, E.J., & BEGEMAN, M.L. (1989). gIBIS: A Tool for All Reasons. *Journal of the American Society for Information Science* 40, 3, 200-213.
- CONKLIN, J. (1987). Hypertext: An Introduction and Survey. *IEEE Computer* 20(9), 17-41.
- COLLIER, G. H. (1987). Thoth-II: Hypertext with Explicit Semantics. Hypertext '87 Proceedings, 269-290.
- DAVIS, H.C. (1995a). To Embed or Not to Embed. *Communications of the ACM*, 38(8), 108-109.
- DAVIS, H.C. (1995b). Data Integrity Problems in an Open Hypermedia Link Service. Ph.D. Dissertation, University of Southampton.
- DAVIS H.C., KNIGHT S. & HALL W. (1994). Light Hypermedia Link Services: A Study of Third-Party Application Integration. Proceedings of the Sixth ACM Conference on Hypertext, ACM Press, New York, pp. 41-50.
- DAVIS, H.C., HALL, W., HEATH, I., HILL, G. & WILKINS, R. (1992). Towards an Integrated Information Environment with Open Hypermedia Systems. in: Proceedings of the ACM Conference on Hypertext, Milan, 181-190.
- DEROSE, S.J. (1989). Expanding the Notion of Links. in Proceedings of the 1989 Hypertext Conference, Pittsburgh, 249-257.
- ENGLEBART, D.C. & ENGLISH, W. (1968). A Research Center for Augmenting Human Intellect. Proceedings of the Fall Joint Computer Conference, Arlington, 395-410.
- GARZOTTO, F., MAINETTI, L. & PAOLINI, P. (1996). Navigation in Hypermedia Applications: Modelling and Semantics. *Journal of Organizational Computing*, Nov. 1996.
- HALASZ, F., & SCHWARTZ, M. (1994). The Dexter hypertext reference model. *Communications of the ACM*, 37(2), 30-39.
- ISAKOWITZ, T., STOHR, E., & BALASUBRAMANIAN, P. (1995). RMM: A Methodology for Structuring Hypermedia Design. *Communications of the ACM* 38(8), 34-44.
- JOYCE, M. (1991). Storyspace as a hypertext system for writers and readers of varying ability. Hypertext'91 Proceedings, ACM, pp 381-388.
- KACMAR, C. (1996). A Process Approach to Providing Hypermedia Services to Existing Non-Hypermedia Applications. *Journal of Electronic Publishing: Organization, Dissemination and Design* (forthcoming).
- MAIOLI, C., SOLA, S. & VITALI, F. (1993). Wide-Area Distribution Issues in Hypertext Systems. ACM SIGDOC'93 Conference Proceedings, ACM.
- MARSHALL, C.C. & IRISH, P.M. (1989). Guided Tours and On-Line Presentations: How Authors Make Existing Hypertext Intelligible for Readers. Hypertext '89 Proceedings, 15-42.
- MARSHALL, C.C., HALASZ, F. G., ROGERS, R.A. & JANSSEN, W.C. JR. (1991). Aquanet: a hypertext tool to hold your knowledge in place. Hypertext'91 Proceedings, 261-275.
- MARSHALL, C.C. & SHIPMAN, F. III (1995). Spatial Hypertext: Designing for Change. *Communications of the ACM* 38(8), 88-97.
- MEYROWITZ, N. (1989). Hypertext - Does it Reduce Cholesterol, Too?, Hypertext '89 Keynote Address, IRIS Technical Report 89-9
- MONNARD, J. & PASQUIER-BOLTUCK, J. (1992). An Object-Oriented Scripting Environment for the WEBSS Electronic Book System. Proceedings of the Second European Conference on Hypertext, ACM, 81-90.
- MYLONAS, E. & HEATH, S. (1990). Hypertext from the Data Point of View: Paths and Links in the Perseus Project. ECHT'90, 324-336.
- NELSON, T.H. (1987). *Literary Machines*. 87.1, Sausalito Press.

- NELSON, T.H. (1995). The Heart of Connection: Hypermedia Unified by Transclusion. *Communications of the ACM*, 38(8), 31-33.
- NIELSEN, J. (1990). The Art of Navigating through Hypertext. *Communications of the ACM* 33(3), 296-310.
- NANARD, J. & NANARD, M. (1991). Using Structured Types to Incorporate Knowledge into Hypertext. In: *Hypertext '91 Proceedings*, 329-343.
- NANARD, J. & NANARD, M. (1993). Should Anchors Be Typed Too? An Experiment with MacWeb. In *Hypertext '93 Proceedings*, 51-62.
- NANARD, J. & NANARD, M. (1995). Hypertext Design Environments and the Hypertext Design Process. *Communications of the ACM* 38(8), 49-56.
- OINAS-KUKKONEN, H. (1997). Embedding Hypermedia into Information Systems, *HICSS'97*.
- PARUNAK, H. V. D. (1991). Ordering the Information Graph. in *Hypertext/Hypermedia Handbook*, BERK, E. & DEVLIN, J. (eds.), Intertext Publications/McGraw-Hill Publishing Co., Inc., New York, 299-325.
- PARUNAK, H. V. D. (1989). Hypermedia Topologies and User Navigation. in *Proceedings of the 1989 Hypertext Conference*, Pittsburgh, 43-50.
- RADA, R. (1990). Hypertext Writing and Document Reuse: the Role of a Semantic Net. *Electronic Publishing*, 3, 3-13.
- RIZK, A. & SAUTER, L. (1992). Multicard: An Open Hypermedia System. *Proceedings of the Second European Conference on Hypertext*, ACM, 4-10.
- RAO, U., & TUROFF, M. (1990). Hypertext Functionality: A Theoretical Framework. *International Journal of Human-Computer Interaction*, 4(2), 333-358.
- SCACCHI, W. (1989). On the Power of Domain-Specific Hypertext Environments. *Journal of the American Society for Information Science*, 40 3, 183-191.
- SCHLOSS, R. J. (1996) Novel Business Uses of Independently Created Hyperlinks in the World Wide Web: Basic Mechanism and Examples. *Proceedings of Twenty-Ninth Annual Hawaii International Conference on System Science (HICSS)*, Maui, Volume II, 137-146.
- SCHWABE, D., ROSSI, G. & BARBOSA, S. D. J. (1996). Systematic Hypermedia Application Design with OOHDM. *Hypertext'96 Proceedings*, 116-128.
- SHACKELFORD, D.E., SMITH, J.B. & SMITH, F.D. (1993). The Architecture and Implementation of a Distributed Hypermedia Storage System. *Hypertext '93 Proceedings*, Seattle, 1-13.
- STOTTS, P.D. & FURUTA, R. (1989). Petri-net-based Hypertext: Document Structure with Browsing Semantics. *ACM Transactions on Information Systems*, 7, 1, 3-29.
- STREITZ, N., HAAKE, J., HANNEMANN, J., LEMKE, A., SCHULER, W., SCHÜTT, H. & THÜRING, M. (1992). SEPIA: A Cooperative Hypermedia Authoring Environment. *Proceedings of the ACM Conference on Hypertext*, Milan, 11-22.
- TERVONEN, I., KEROLA, P. & OINAS-KUKKONEN, H. (1997). An Organizational Memory for Quality-based Software Design and Inspection: A Collaborative Multiview Approach with Hyperlinking Capabilities, *HICSS'97*.
- THÜRING, M., HANNEMANN, J. & HAAKE, J. M. (1995). Designing for Comprehension: A Cognitive Approach to Hypermedia Development. *Communications of the ACM*, 38(8), 57-66.
- TRIGG, R. H. (1988). Guided Tours and Tabletops: Tools for Communicating in a Hypertext Environment. *ACM Transactions on Office Information Systems* 6(4), 398-414.
- TRIGG, R.H. & WEISER, M. (1986). TextNet: A Network-Based Approach to Text Handling. *ACM Transactions on Office Information Systems*, 4, 1,1-23.
- TUROFF, M. (1991). Computer mediated communication requirements for group support. *Journal of Organizational Computing*, 1 1, 85-113.
- UTTING, K. & YANKOLOVICH, N. (1989). Context and Orientation in Hypermedia Networks. *ACM Transactions on Information Systems*, 7, 1,58-84.
- VITALI, F. & DURAND D. (1995). Using Versioning to Provide Collaboration on the WWW. *Proceedings of the 4th International World Wide Web Conference*, Boston.
- WANG, W. & RADA, R. (1995). Experiences with Semantic Net Based Hypermedia. *International Journal of Human-Computer Studies*, 43, 419-439.
- WURMAN, R.S. (1989). *Information Anxiety*. Doubleday, NY.
- YODER, E., AKSCYN, R. & MCCracken, D. (1989). Collaboration in KMS, A shared hypermedia system. *Proceedings of the CHI'89 Conference*, Austin, ACM, 37-42.
- ZELLWEGER, P. (1989). Scripted Documents: A Hypermedia Path Mechanism. in *Hypertext '89 Proceedings*, Pittsburgh, 1-14.

## References on the World Wide Web

- DEPARTMENT OF INFORMATION SYSTEMS (1996). Department of Information Systems home page. <http://is-2.stern.nyu.edu/isweb>
- GRAMLICH, W.C. (1994). Public Annotation Systems. <http://playground.sun.com:80/~gramlich/1994/annote/>
- LA LIBERTE, D. (1994). A Protocol for Scalable Group and Public Annotations. <http://union.ncsa.uiuc.edu/~liberte/www/scalable-annotations.html>
- MALONEY, M. & QUIN, L. (1996). Hypertext links in HTML. Internet Draft. Available as <ftp://ftp.ietf.cnri.reston.va.us/internet-drafts/draft-ietf-html-relrev-00.txt>
- NATIONAL CENTRE FOR SUPERCOMPUTING APPLICATIONS (UNIVERSITY OF ILLINOIS) (1995). The NCSA Mosaic Homepage. <http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html>
- SIGLINK (1994). SIGLINK Home Page, <http://www.acm.org/siglink/>
- VANZYL, A. J., CESNIK, B., HEATH, I. & DAVIS, H.C. (1994). Open Hypertext Systems : An Examination of Requirements, and Analysis of Implementation Strategies, comparing Microcosm, HyperTED, and the World Wide Web. <http://elecpress.lib.monash.edu.au/papers/openhypermedia.html>