Hypermedia on the Web: What Will It Take?

Fabio Vitali

University of Bologna
Department of Computer Science   *Web: http://www.cs.unibo.it/*
Mura A. Zamboni, 7
I-40127 Bologna BO Italy
Email: fabio@cs.unibo.it

Michael Bieber

New Jersey Institute of Technology
Hypermedia Research Laboratory
Computer and Information Science Department
University Heights, Newark, New Jersey 07102 U.S.A.
Email: bieber@njit.edu
Web: http://www-ec.njit.edu/~bieber

Abstract: Researchers in the hypermedia field often lament that the World Wide Web does not support many of hypermedia's rich structuring, navigation and annotation features. What would it take for everyday Web applications to be fully hypermedia compliant, now that the basic hypermedia building blocks exist on the Web? The following four capabilities are the most critical for integrating hypermedia support in the Web environment: edit-capable browsers, storing document content and link anchors separately, external linkbases, and displaying link spans, node and link attributes. Individual developers cannot decide autonomously on how to resolve many of the outstanding issues. Developers need agreed-upon conventions and tools built upon today's Web standards to fully incorporate hypermedia functionality into everyday applications.

Categories and Subject Descriptors: H.5.4. **[Information interfaces and presentation]** Hypertext/Hypermedia - *Architectures*

General Terms: standardization, design

Additional Key Words and Phrases: hypertext, hypermedia, World Wide Web, browsers, linkbases, link attributes, hypertext functionality

Introduction

Researchers in the hypermedia field often lament that the World Wide Web does not support many of hypermedia's rich structuring, navigation and annotation features [Bieber 1997], [Bieber 1997a] developed over the last 40 years since Engelbart's NLS system Engelbart 1968]. Structuring features include semantically typed anchors, links and nodes; attributes and keywords on these; links to and from content spans within any medium; bidirectional links;

links recursively to links and to other hypermedia constructs; transclusions and inclusions providing access to all uses of the same span of content; versioning of hypermedia constructs; landmarks; automatic link propogation; trails and guided tours; and local and global overviews. Navigation features include link traversal; visual feedback upon link arrival; search based on the hypermedia structure (as opposed to content); process enactment through link traversal; backtracking strategies; backjumping; and history mechanisms. Annotation features include comments, bookmarks and reader-authored links with private, workgroup and public access. People can annotate any hypermedia construct, including other annotations recursively.

In fact, standard Web data formats and protocols recently introduced can facilitate many of these. So what is really standing in the way? What would it take for everyday Web applications to be fully hypermedia compliant? Now that the basic hypermedia building blocks exist, hypermedia and WWW researchers must now figure out how to deploy them to create an infrastructure for seamlessly integrating hypermedia techniques into the Web environment and making them pervasive in everyone's daily Web activities.

We believe that the following four capabilities are the most critical for integrating hypermedia support to the Web environment:

- edit-capable browsers
- storing document content and link anchors separately
- external linkbases
- displaying link spans, node and link attributes

Edit-capable browsers are, in part, a human-computer interaction issue; people want to work collaboratively in a single, seamless environment without switching to separate authoring tools. But more fundamentally, hypermedia advocates believe strongly that readers should be able to author at will, with the ability to add links and annotations to any document, and this must take place in the browser as the user reads. (Furthermore, these browsers should support the creation of hypertext constructs such as semantic node and link types and other attributes, trails and guided tours, etc.) Also, browsers should automatically display destination link spans, and metainformation such as semantic types, keywords and other attributes as part of maintaining the user's orientation.

Hypermedia features such as bi-directional linking, personalized links and structural search rely on links as "first class objects" maintained external to documents instead of embedded inside anchors. Furthermore, ubiquitous linking from documents not belonging to the user requires links and anchors be maintained external to documents and only merged as the document is sent for display. This requires external linkbases and anchor points maintained outside documents.

Edit-capable Browsers

Most pre-Web hypermedia systems [Conklin 1987a], as well as the first WWW browser designed by Tim Berners-Lee, allowed editing and link traversal. The first widely used Web browser, Mosaic, dropped authoring and established the Web as a hypermedia system viewed for publishing to large audiences. Today's edit-capable browsers (e.g., AOLpress, FrontPage, Netscape Composer, as well as professional site builders such as Adobe GoLive and Macromedia DreamWeaver) make this task somewhat easier, but the Web still is not viewed by most as a personal workspace in the way Microsoft Office is. Furthermore, today's edit-capable browsers do not support collaborative authoring, which few off-web environments do either. But people need this, and the Web can provide the networked infrastructure for collaborative work. Again, some pre-Web hypermedia systems did support collaborative authoring [Streitz 1992], [Smith 1991], [Conklin 1987a].

WebDAV [WebDAV 1999] is an IETF standard extending the HTTP protocol to allow write access to the content of a Web site. This would allow Web applications to be able to read as well as write content on a Web site, and to integrate the navigation and the authoring in the same tool.

WebDAV is strong enough to support complex, multi-authored projects including versioning, link management and referential integrity. Yet the danger is that developers will use WebDAV to build just a generic readable and writeable networked file-system for all kinds of data (including hypermedia data), but not explicitly support hypermedia constructs. This would make it harder to provide sophisticated collaborative hypermedia functionalities such as notification, fine-grained locking, lock-free collaboration, and support for hypermedia transclusions (see for instance [Nelson 1987] and [Nelson 1999] and [Wiil 1993a]). Developers must be pushed to develop these features.

Storing document content and link anchors separately

Unless an HTML document is dynamically generated, anyone without write-access can neither create anchors at a link destination within that document, nor create links leading from that document. Several interesting hypermedia features remain impossible until we can extract link anchors from document content, including bi-directional linking (links that can be activated at both ends, and thus not providing a preferred direction of navigation) and personalized links (visible to a group of users, rather than all readers of a document). About this problem, see also [Brailsford 1999].

Two issues arise when separating content and links. The first is the problem of locators, i.e., of unequivocally identifying the exact spot in the document at which link anchors should be attached. Of course, this is extremely media type-dependent; methods to specify a location within ASCII documents, marked-up texts or, say, bitmaps will vary considerably. The second issue is keeping the locators correctly pointing to the right positions when either or both linked documents are being modified asynchronously.

XPointer [Maler 1999] is an ancillary standard to XML, the new markup language being proposed for the World Wide Web. An XPointer is an elaborate specification of a location within a text or XML document, by way of expressing the steps necessary to get from well-known document elements (e.g., the start, the named elements, the destination of the preceding link) to the required destination in terms of tree-traversal steps, character counting, and other methods. XPointer has yet to be widely accepted. The many reliability and stability problems necessarily not yet dealt with in its proposal, are important issues for XPointers. Because their sophistication and precision makes them extremely fragile, any element of a possibly complex chain of specifications within an XPointer could be outdated by even the simplest change in the document it refers to.

Davis ([Davis 1998], but see also [Davis 1999] in this issue) discusses four solutions to automatically resolving this referential integrity of links problem:

1. Don't bother: let the user find the correct destination autonomously
2. Provide an off-line batch link verifier that identifies dangling locators at night, and asks the hypermedia owner to update them to the correct position.
3. Provide an opportunistic, on-demand heuristic engine that scans through a document just before displaying it, in order to determine the dangling locations and the best guesses for their current positions.
4. Require all documents to be versioned, and deduce the current positions of dangling locators by examining the differences between the version of the document in which the locators were created and the current one. This could be obtained either by creating and comparing a "diff" between the two versions, or by employing a sufficiently fine-grained delta format, clearly listing document changes.

No solution is without drawbacks, and yet none is inherently flawed. The appropriateness of each depends on the kind of application and link type. For instance, the precision required in correctly identifying a transcluded portion of a document necessitates the reliability of a full versioning system.

No one developer can decide autonomously on such important decisions for the Web as a whole. Instead additional standards and agreements are necessary for these features to actually work and interoperate.

External Linkbases

Once we have a reliable separate addressing mechanism for point-to-point links, how can we actually create external links? How do we express them? How can we store them? How is the browser supposed to know about their existence, so that it can merge and display them with the documents they connect?

In the hypermedia literature, the notion of a linkbase has often appeared (e.g., [Pearl 1989]), especially in Open Hypermedia Systems research systems [OHS 1999], [Nürnberg 1998]. Links are stored outside of the documents they refer to, in one or more databases of links that are queried by the browser just before showing the document. This architectural choice is extremely flexible and powerful, making many currently unavailable functionalities possible. For instance, this enables different link sets over the same content for different audiences and tasks [Carr 1998], [Lewis 1999]. Third-party individuals can provide links and create guided tours over documents owned by others. Individuals and groups could maintain their own personal linkbases. In a way, this functionality can be had in the current Web, by creating appropriate framesets with HTML 4.0. Although technically possible, this solution is clearly a sophisticated hack that is hard to generalise.

The XLink [DeRose 1999], [DeRose 2000] standard provides the syntax within XML to express both internal links (similar to the "A" element in HTML) and external links, where both (or all, in case of multi-ended links) of the end-points are stored outside of the documents they refer to and use XPointers to identify their exact location. Thus XLinks can be stored in XML documents (de facto linkbases) that are independent of the documents they actually connect. An additional nice feature of XLinks is that although they are expressed with XML syntax, it is possible to specify links over any kind of document for which an XPointer can be expressed.

What the XLink proposal does not cover is how to use these external links. How do we associate a link document to a content document? How can we provide for additional or alternate link documents to the "official" one? How can we express the fact that independently of what is explicitly expressed, an additional, fixed set of link documents containing the private links of the reader should be added to all visited documents? These problems are similar to and yet more complex than the ones stylesheets present, since they involve multiple link sources and linkbase owners, in a situation much more complex than CSS stylesheets often handle.

Here too, individual developers can not decide autonomously on how to resolve these issues for the full Web environment. More research and agreements are needed for external link bases to become a standard feature of our daily work.

Displaying link spans, and node and link attributes

Link anchors represent a target area or scope of the link's underlying relationship within each endpoint node. Traversing a link should show the relationship's scope in the destination node. For example, if the link connects a document with a paragraph contradicting a paragraph in another document, then traversing the link should highlight the contradicting paragraph. The HTML language allows node-to-node, span-to-node, and span-to-span links through the "LINK" element using a single "A" element or a pair of "A" elements. However, because today's browsers scroll to destination points instead of highlighting destination spans, few bother to even mark destination spans in a document.

Attributes include semantic types, keywords and other metainformation and attributes for both nodes and links. Browsers should have a mechanism built in to display these automatically, perhaps when the user moves the mouse over an anchor (for link metainformation) or the document title bar (for node metainformation).

The RDF proposal by the W3C [Lassila 1999] addresses metainformation within Web documents. RDF can record both node and link metainformation, though proposed standard sets of attributes often only describe node metainformation. Rather than a predefined set of fields and permitted values, RDF provides a general syntax for expressing metainformation, and leaves the task of specifying appropriate sets to further conventions (for instance, the Dublin Core set [DublinCore 1999]). Thus we are two steps away from being able to enjoy node and link attributes on the Web. Once the RDF standard is in place, a standard set of metainformation fields must be agreed upon, and then browsers must support standard ways to add them, display them, and otherwise use them (e.g., in structural search). While the Dublin Core Set seems a likely standard set of node attributes, no proposal has been put forth for an architecture that uses them in Web environments.

Until we do provide a seamless, integrated environment for hypermedia and our own daily activities on the Web, few application developers and users will have the incentive, knowledge and energy required to deploy hypermedia functionality [Bieber 1999b], [Bieber 1997]. The Web finally has the standard data formats and protocols. We now need to provide the tools and conventions so that every user can benefit from them.

Acknowledgements

We gratefully acknowledge funding for this line of research by the NASA JOVE faculty fellowship program, by the New Jersey Center for Multimedia Research, by the National Center for Transportation and Industrial Productivity at the New Jersey Institute of Technology (NJIT), by the New Jersey Department of Transportation, and by the New Jersey Commission of Science and Technology, and by Rutgers University.

References

[Bieber 1997] Michael Bieber and Fabio Vitali. "Toward Support for Hypermedia on the World Wide Web" in IEEE Computer, 30(1), 62-70, 1997.

[Bieber 1997a] Michael Bieber, Fabio Vitali, Helen Ashman, V. Balasubramanian and Harri Oinas-Kukkonen. "Fourth Generation Hypermedia: Some Missing Links for the World Wide Web" in International Journal of Human Computer Studies, 47 (1), 31-65 [Online: http://www.hbuk.co.uk/ap/ijhcs/webusability/], July 1997.

[Bieber 1999b] Michael Bieber, Harri Oinas-Kukkonen, and V. Balasubramanian. "Hypertext Functionality" in ACM Computing Surveys, Symposium on Hypertext and Hypermedia, 1999.

[Brailsford 1999] Brailsford, D.F., "Separable hyperstructure and delayed link binding" in ACM Computing Surveys, Symposium on Hypertext and Hypermedia, 1999.

[Carr 1998] Leslie A. Carr, Wendy Hall, and Steven Hitchcock. "Link Services or Link Agents?" in Proceedings of ACM Hypertext '98, Pittsburgh PA, 113-122, [Online: http://acm.org/pubs/citations/proceedings/hypertext/276627/p113-carr/], June 1998.

[Conklin 1987a] Jeff Conklin. "Hypertext: An Introduction and Survey" in IEEE Computer, 20(9), 17-41, September 1987.

[Davis 1998] Hugh C. Davis. "Referential Integrity of Links in Open Hypermedia Systems" in Proceedings of ACM Hypertext '98 , Pittsburgh, PA, 207-216, June 1998.

[Davis 1999] Hugh C. Davis. "Hypertext Link Integrity" in ACM Computing Surveys, Symposium on Hypertext and Hypermedia, 1999.

[DeRose 1999] Steven J. DeRose. "XML Linking" in ACM Computing Surveys, Symposium on Hypertext and Hypermedia, 1999.

[DeRose 2000] Steven DeRose, Eve Maler, David Orchard, and Ben Trafford (editors). XML Linking Language (XLink), World Wide Web Consortium Working Draft3 [Online: http://www.w3.org/TR/WD-xlink], January 2000.

[DublinCore 1999] DublinCore, http://purl.org/DC, 1999.

[Engelbart 1968] Douglas C. Engelbart and William K. English, "A Research Center for Augmenting Human Intellect" in Proceedings of the Fall Joint Computer Conference (IFJC), 33, Arlington, VA, pages 395-410, 1968.

[Lassila 1999] Ora Lassila and Ralph Swick (editors), "Resource Description Framework (RDF) Model and Syntax Specification" World Wide Web Consortium Recommendation, [Online: http://www.w3.org/TR/REC-rdf-syntax/], February 22 1999.

[Lewis 1999] Paul H. Lewis, Wendy Hall, L. Carr, and David DeRoure. "The Significance of Linking" in ACM Computing Surveys, Symposium on Hypertext and Hypermedia, 1999.

[Maler 1999] Eve Maler and Steven DeRose (editors). XML Pointer language (XPointer), World Wide Web Consortium Working Draft, [Online: http://www.w3.org/TR/WD-xptr], 1999.

[Nürnberg 1998] Peter J. Nürnberg, John J. Leggett, and Uffe K. Wiil, "An Agenda for Open Hypermedia Research" in Proceedings of ACM Hypertext '98, Pittsburgh, PA, 198-206, [Online: http://acm.org/pubs/citations/proceedings/hypertext/276627/p198-nurnberg/], June 1998.

[Nelson 1987] Theodor Helm Nelson. Literary Machines, Edition 87.1, Sausalito Press, 1987.

[Nelson 1999] Theodor Helm Nelson. "Xanalogical Media: Needed Now More than Ever" in ACM Computing Surveys, Symposium on Hypertext and Hypermedia, 1999.

[OHS 1999] OHS, http://www.csdl.tamu.edu/ohs/, 1999.

[Pearl 1989] Amy Pearl. "Sun's Link Service: A Protocol for Open Linking" in Proceedings of ACM Hypertext '89, Pittsburgh, PA, 137-146, November 1989.

[Smith 1991] John B. Smith and F. Donelson Smith. "ABC: A Hypermedia System for Artifact-Based Collaboration" in Proceedings of ACM Hypertext '91, San Antonio, TX, 179-192, December 1991.

[Streitz 1992] Norbert A. Streitz, Jörg M. Haake, Jörg Hannemann, Andreas Lemke, Wolfgang Schuler, Helge Schütt and Manfred Thüring. "SEPIA: A Cooperative Hypermedia Authoring Environment" in Proceedings of the ACM Conference on Hypertext (ECHT '92), Milano, Italy, 11-22, December 1992.

[WebDAV 1999] WebDAV, http://ww w.ietf.org/html.charters/webdav-charter.html, 1999.

[Wiil 1993a] Uffe K. Wiil and John J. Leggett . "Concurrency Control in Collaborative Hypertext Systems" in Proceedings of ACM Hypertext '93, Seattle, WA, 14-24, November 1993.