

UC Irvine

ICS Technical Reports

Title

Final report of the San Mateo Meeting of the Working Group on Distributed Authoring on the World Wide Web : America Online Productions, San Mateo, CA, July 10, 1996

Permalink

<https://escholarship.org/uc/item/8z5502pk>

Author

Whitehead, E. James, Jr.

Publication Date

1996-08-09

Peer reviewed

ARC
Z
699
03
no. 96-32

**Final Report of the San Mateo Meeting of the
Working Group on Distributed Authoring on the
World Wide Web**

**America Online Productions, San Mateo, CA
July 10, 1996**

Edited by E. James Whitehead, Jr.

UCI-ICS Technical Report 96-32
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425

August 9, 1996

Notice: This Material
may be protected
by Copyright Law
(Title 17 U.S.C.)

Table of Contents

TABLE OF CONTENTS.....	1
INTRODUCTION.....	2
EXECUTIVE SUMMARY.....	3
MEETING PARTICIPANTS.....	4
FINAL AGENDA.....	5
MINUTES.....	6
MORNING (9:00 AM).....	6
<i>History of Distributed Web Authoring - Whitehead</i>	6
<i>Working Group Descriptions and Purposes - Whitehead</i>	7
<i>Collaborative Authoring in Microsoft Word - Fein</i>	8
<i>Distributed Authoring - Nielsen</i>	12
AFTERNOON (1:35 PM).....	14
<i>AOLpress and AOLserver - Long</i>	14
<i>FrontPage - Schulert</i>	17
<i>Goal of Working Group - Whitehead</i>	18
<i>Sponsorship of Working Group by World Wide Web Consortium</i>	19
<i>Meeting Goals, Criteria for Completion - Whitehead</i>	20
<i>Key Interoperability Issues (Nielsen's List)</i>	21
<i>Specified Deliverables</i>	22
SLIDES: JIM WHITEHEAD (AGENDA, BACKGROUND, DISCUSSION).....	24
SLIDES: RON FEIN (COLLABORATIVE AUTHORIZING IN MICROSOFT WORD)...	35
SLIDES: HENRIK FRYSTYK NIELSEN (DISTRIBUTED AUTHORIZING).....	45
SLIDES: DAVE LONG (AOLPRESS/AOLSERVER).....	50

Introduction

The first meeting of the Working Group on Distributed Authoring on the World Wide Web was held on July, 10, 1996, at the offices of America Online (AOL) Productions, 2600 Campus Drive, San Mateo, California, USA. The main objective of this meeting was to provide a forum where people interested in fostering interoperability between distributed web authoring applications could meet, exchange information about the current state of the art and practice, identify key interoperability issues, and formulate an agenda for achieving interoperability.

While there are many tools on the market (e.g., SoftQuad HotMetal Pro 2.0, Quarterdeck WebAuthor 2.0, Adobe PageMill 1.0, InContext Spider 1.1) which provide a graphical user interface for the authoring of HyperText Markup Language (HTML) format files and layout of associated graphic content, these tools require the user to have direct access to the physical storage of the name space served by a HyperText Transfer Protocol (HTTP) server. This effectively limits the use of these tools to computers connected to a local area network (LAN) containing the physical storage. In contrast, an emerging class of tools (e.g., Microsoft FrontPage 1.1, AOLpress/AOLserver 1.1) allow users to save their work directly to an HTTP server, affording a style of work where authors are located remotely to the HTTP server hosting their content. This second class of tools, known as distributed web content authoring applications, were the focus of this working group.

Meeting participants included members from the IETF HTTP Working Group, the World Wide Web Consortium, the document management community, software configuration management companies, distributed web authoring tool vendors and researchers, and the academic hypertext versioning community. This wide range of expertise resulted in very detailed and considered discussion of issues concerning distributed web content authoring.

This document contains:

- An executive summary of the meeting
- A list of participants at the meeting
- Final agenda for the meeting
- Minutes of the meeting
- Copies of transparencies shown at the meeting

This meeting material is also available at the meeting home page:

<http://www.ics.uci.edu/~ejw/authoring/sanmateo/>

I (the organizer) would like to thank Dave Long of America Online for arranging the conference room at AOL Productions, and for his in-meeting support. I would also like to thank Ron Fein, Henrik Frystyk Nielsen, Dave Long, and Andy Schulert for their insightful presentations, Dennis Hamilton and Keith Dawson for their excellent notes, and all participants for their thoughtful discussion.

Irvine, California, August 9, 1996

Jim Whitehead

Executive Summary

On July, 10, 1996, an informal meeting of the Working Group on Distributed Authoring on the World Wide Web (WWW) was held at the offices of America Online (AOL) Productions in San Mateo, California. There were 14 meeting participants, including members of the Internet Engineering Task Force (IETF) HyperText Transfer Protocol (HTTP) Working Group, the World Wide Web Consortium (W3C), the document management community, software configuration management companies, distributed web authoring tool vendors and researchers, and the academic hypertext versioning community. The main objective of this meeting was to provide a forum where people interested in fostering interoperability between distributed web authoring applications could meet, exchange information about the current state of the art and practice, identify key interoperability issues, and formulate an agenda for achieving interoperability.

The meeting began with a presentation by Jim Whitehead, U.C. Irvine on the history of distributed authoring on the WWW, on the purpose of the distributed authoring working group, and on the purpose of the Working Group on Versioning and Configuration Management of World Wide Web Content. Ron Fein, Microsoft, next gave a presentation on collaborative authoring and WWW features in Microsoft Word. A presentation by Henrik Frystyk Nielsen, W3C, on distributed authoring and his vision of a unified PUT model, finished the morning session. The afternoon began with a presentation by Dave Long, America Online, on AOLpress and AOLserver, which was followed by a presentation by Andy Schultert, Microsoft, on FrontPage. Whitehead next led discussion on the goal of the working group, the sponsorship of the working group, and criteria for completion. At the end of the day, the group discussed what they considered to be key interoperability issues, and developed a list of activities and deliverables to be produced by the working group.

Key Results

1. The working group adopted as its goal: Enable distributed web authoring tools to be broadly interoperable.
2. Create a task-oriented list of scenarios which interoperable distributed authoring tools should be able to perform. Keith Dawson, Atria, is the editor of this document.
3. Create a list which collates the "key functionality" among AOLpress/AOLserver, FrontPage, Word, as well as other distributed authoring tools. Dave Long, America Online, is the editor of this document.

The working group also identified some areas for future work:

1. Criteria for completion of the group's work were discussed, and need to be clearly described in writing.
2. The working group was tentatively interested in seeking sponsorship by the W3C. However, before the group is officially sponsored, members need to know the intellectual property rights implications of such sponsorship.
3. There should be a reinvestigation of work previously performed by Murray Maloney on standard LINK REL and REV tags.
4. There should be an investigation by the versioning and configuration management group into the utility of entity tags (etags) for performing versioning.

Detailed meeting notes, a list of participants, a final agenda for the meeting, and slides from the meeting are available at <http://www.ics.uci.edu/~ejw/authoring/sanmateo/>.

Meeting Participants

Bruce Brown	America Online, Santa Barbara	<i>Bruce@sb.aol.com</i>
Ed Burns	Silicon Graphics	<i>edburns@sgi.com</i>
Keith Dawson	Atria Software	<i>dawson@atria.com</i>
Lee Farrell	Canon Information Systems	<i>lfarrell@cisoc.canon.com</i>
Ron Fein	Microsoft	<i>ronfe@microsoft.com</i>
Dennis E. Hamilton	Xerox PARC	<i>hamilton@parc.xerox.com</i>
Dave Long	America Online, Santa Barbara	<i>dave@sb.aol.com</i>
Larry Masinter	Xerox PARC	<i>masinter@parc.xerox.com</i>
Henrik Frystyk Nielsen	World Wide Web Consortium	<i>frystyk@w3.org</i>
Kenji Ota	NTT Software Laboratories	<i>ota@nttlabs.com</i>
Christopher Seiwald	P3 Software	<i>seiwald@p3.com</i>
Andrew Schulert	Microsoft	<i>andyschu@microsoft.com</i>
Jim Whitehead	Univ. of California, Irvine	<i>ejw@ics.uci.edu</i>
Tom Wills	Xerox XSoft	<i>twills@xsoft.xerox.com</i>

Final Agenda

Morning (9:00 AM)

- Introductions
- Overview of efforts to date on distributed authoring
- Discussion of the direction and purpose of the working group
- Review of current distributed web authoring systems and the mechanisms they employ:
 - ✦ Ron Fein on collaborative web authoring in Microsoft Word
 - ✦ Henrik Frystyk on Amaya, PUT model
 - ✦ Dave Long on GNNpress/GNNserver
 - ✦ Andy Schulert on FrontPage

Afternoon (1:30 PM)

- Identification of key areas in which existing systems are unable to interoperate
- Discussion of which key areas can be profitably addressed by this working group, and a rough prioritization of the areas
- Solicitation of volunteers for coordinating work on improving interoperability in key areas
- Development of criteria for evaluating the success of key area work groups
- Establishing timelines for action

Notes

As sometimes happens, aggressive schedules slip. In the case of this meeting, Dave Long and Andy Schulert actually gave their presentations after lunch. This resulted in a more compressed time for discussion. The topic of the working group's purpose and sponsorship was revisited in the afternoon session, immediately after the talks.

Minutes

Editor's note: These minutes are the result of merging notes taken by Dennis Hamilton, Keith Dawson, and Jim Whitehead. In particular, the detailed, comprehensive notes taken by Dennis Hamilton form the core of this material. These original notes were edited to make them more readable in a standalone document, and to add context so that people who did not attend the meeting could still understand the discussion. Thus, text attributed to individuals should not be interpreted as a direct transcription of what they said in the meeting, but rather as a representation of their content or position, consistent with what was actually spoken.

Morning (9:00 AM)

The meeting began with all present introducing themselves, and stating their organization.

History of Distributed Web Authoring - Whitehead

Whitehead gave a presentation on the history of distributed web authoring. The first web authoring tool was the Nexus browser (1989/1990), the Next machine browser/editor developed by Tim Berners-Lee. In 1992, the Mosaic 2.4 browser achieved critical mass, and created the "publish/browse" technical frame of reference for the WWW. In 1992, the HTTP PUT method disappeared from the HTTP 1.0 specification because this specification was intended to reflect the practices of existing HTTP servers, which did not have write capability. In 1994/95, development took place on NaviPress/NaviServer (now AOLpress/AOLserver), Vermeer (now Microsoft) FrontPage, and the World Wide Web Consortium (W3C) Line Mode Browser, the first generation of distributed web authoring tools.

In December, 1994, there was a breakout session on distributed web authoring tools at the WWW4 conference. The focus of this session was how to achieve interoperability among distributed web authoring tools. This session identified the following interoperability issues: common access control model, "lost update" problem, need for BROWSE and MKDIR HTTP methods, editing of variants, access to "raw" HTML before server-side include (SSI) processing, strong authentication, and placing a link to the HTML Standard Generalized Markup Language (SGML) Document Type Description (DTD) in the HTML source.

In 1995/96, Rohit Khare at W3C researched a resource leasing and locking mechanism. In March 1996, Dan Connolly of the W3C put out a call for volunteers to coordinate distributed authoring activity, and Jim Whitehead of U.C. Irvine volunteered, forming the Working Group on Distributed Authoring on the World Wide Web, and the Working Group on Versioning and Configuration Management of World Wide Web Content. Though Whitehead received much appreciated assistance from Dan Connolly of the W3C, and Larry Masinter of the IETF HTTP Working Group, neither the distributed authoring or the versioning and configuration management group has official sponsorship from either the W3C or the IETF.

In June, 1996, the IETF HTTP Working Group completed an internet draft of HTTP version 1.1, including a PUT (write) method. In July 1996, the IETF HTTP working group completed an internet draft on digest authentication for HTTP. On July 10, 1996, the Working Group on Distributed Authoring on the World Wide Web met at AOL Productions, San Mateo, California.

Working Group Descriptions and Purposes - Whitehead

Whitehead next gave a presentation stating the best current description of the purpose and activities of the Working Group on Distributed Authoring on the World Wide Web, and the Working Group on Versioning and Configuration Management of World Wide Web Content. Whitehead stated that the mission of the distributed authoring group is to make distributed authoring as pervasive as browsing is today. This could be achieved by specifying preliminary modifications to existing internet specification (e.g. HTTP), and by establishing usage conventions (e.g., URL naming conventions). These preliminary specification and usage conventions would then be forwarded to appropriate bodies, such as the IETF HTTP Working Group, for final discussion and potential incorporation into existing internet standards. (*Further discussion of this issue took place in the afternoon.*)

The Working Group on Distributed Authoring has a home page at URL:

<http://www.ics.uci.edu/~ejw/authoring/>

It also has a mailing list, <w3c-dist-auth@w3.org>, hosted by the World Wide Web Consortium.

At this point there was some discussion about membership on the mailing list. Whitehead stated that he hasn't yet turned down anyone who wanted to join the list, and he wants high participation on the list. However, it is a managed list. Whitehead wants the ability to gag (after warning) list participants who are not behaving constructively. Masinter stated that this is a perilous course, that it is better to just leave the list open and be selective in reading and responding.

There was some discussion about the differences between W3C mailing lists, which do not have to have open membership, and IETF working group mailing lists, which do.

Masinter: Irrespective of whether the mailing list is open or restricted, gagging on an individual basis just doesn't work out. I suggest having solid guidelines about the list and what is expected of participation, and have that known in advance.

Whitehead: We will discuss membership and sponsorship further later today.

Masinter: You need to decide on sponsorship of this working group so that the participants who are concerned about legal questions (e.g., anti-trust and intellectual property rights) around participating with others can be satisfied and operate appropriately.

The Working Group on Versioning and Configuration Management of World Wide Web Content has as its goal the development of a preliminary specification for how to provide versioning and configuration management of content served by an HTTP server. This goal can be achieved through the same means used by the distributed authoring group: extension of existing specifications, and establishment of usage conventions. While the issues considered by the distributed authoring group also include versioning and configuration management, the versioning group was created as a sub-group of the distributed authoring group to partition the issue space due to the complexity and number of issues involved.

The Working Group on Versioning and Configuration Management has a home page at URL:

<http://www.ics.uci.edu/~ejw/versioning/>

It also has a mailing list, <www-vers-wg@ics.uci.edu>, hosted by the Department of Information and Computer Science at the University of California, Irvine.

Seiwald: Is it possible to separate issues of versioning and configuration management from distributed authoring, and how do we reconcile different interests?

Whitehead: There is significant overlap between the two issue spaces. For example, access control is considered by both groups. However, due to the size of the versioning and configuration management issue space, and the different set of parties interested in this issue, such as configuration management companies, it makes sense to have a separate group address this issue. There will be close collaboration between the two groups, and there is currently significant overlap in membership.

Hamilton: The issue spaces aren't orthogonal, but we want to factor these topics (collaboration, versioning, authoring) so that we can proceed in parallel as much as possible.

Seiwald made a comment about configuration management being a result of cooperative authoring and how versioning and authoring don't get too far apart.

Fein asked about previous work on collaborative editing and the WWW.

Nielsen: There has been some discussion about collaboration in other groups, and there was a workshop on the subject.

Masinter: What are people interested in?

At this point we went around the room and reviewed what people were interested in.

Question: Have any of the people from Netscape Navigator Gold been asked to participate in this meeting? Long had some contact but it seems to have gone quiet at his end at the moment. Whitehead mentioned that Netscape was contacted prior to the meeting, but had declined to participate. Before the meeting someone from Netscape did telephone Dan Connolly at the W3C to mention they would be unable to attend the meeting, but were interested in participating in the working group.

There was a brief discussion about including multiple repository searching, etc. by Long, and Nielsen talked about the meeting recently held about indexing, Harvest, etc.

Collaborative Authoring in Microsoft Word - Fein

Fein gave a presentation on collaborative authoring in Microsoft Word, with brief comparisons to other word processors.

Fein mentioned that the efforts of this group are already too late to impact Word 8, which is scheduled to be released this fall for Windows 95, NT and Macintosh.

Fein listed the five most important collaboration features for word processors:

1. Revision Marking
2. Annotations
3. Merging Documents
4. Comparing Documents
5. Access Control

Revision marking slide:

Fein discussed how MS Word and Lotus WordPro both track revisions to a document by storing the revision history in the document file. This has the advantage that moving the document to a floppy doesn't cause the loss of revision information. The desired capability is to know who wrote what, and when they wrote it, at a fine-grain level within the document (i.e., character, word, and paragraph level). It is also desirable to have protection from unauthorized changes by requiring people to submit deltas for approval and selective incorporation. One way of providing this desired capability is to have provisional edits stamped with user/time, which are then incorporated into the document during an accept/reconcile process by the main document author.

Experience with LAN usage shows that the access speed hit for storing the revision history in the document is acceptable. However, Fein suspects this will likely not be the case when retrieving or saving a file at 28.8 baud (the fastest mass-market modem speed).

Experience with Microsoft Internet Explorer has shown that converting a Word file to HTML causes irretrievable loss of revision information. Ideally, it would be nice to have the choice of file format be transparent to the user.

Fein also provided a brief discussion of versioning in Lotus WordPro (not on his slides).

Hamilton: The business of posting revisions or deltas as part of the document file is an approach in OpenDoc.

Burns: The Dynabook DynaWeb server manages deltas within their database, and only delivers the version users want.

There was some discussion about embedding revision information into HTML comments. This has the drawback that it makes the comments encoded. Whitehead mentioned that David Durand and Fabio Vitali are working on a versioned HTML called "VTML."

Annotations slide:

Unlike revisions, annotations are not proposed changes to the document. Within Word, annotations are a separate text stream anchored to regions of the main text. Fein mentioned the desirability for annotations which are not links (i.e., a few lines of text rather than a jump to a new document) to pop up a window within a WWW browser, after a Word file is converted to HTML.

Merging:

Collaborative editing leads to the problem of integration of multiple sets of revisions. Current approach: dump all revisions into the same file, then allow the document editor to choose which revisions to accept. Merging has several implications for the Web. First, merging requires some means of performing revision marking so users can choose between different revisions. Second, until all revisions have been merged, the document is in a half-finished state, and should not be world-readable until the merge has been performed in a rational way.

Comparing versions:

The functionality in Microsoft Word for comparing two different versions of a document was originally developed for the legal market so they could analyze changes made to a contract by a potentially hostile party who will not tell you what changes they have made. This is more difficult than the plain text (e.g. source code) comparison (or 'diff') case, since there is a need to perform differencing on logical parts of a document, such as sentences, paragraphs, and pages, rather than just individual lines of text.

Access control:

The issue here is how to provide fine-grain (individual revisions within a document) permissions to different users. WordPro has very good fine-grain document access control. A difficult access control situation occurs when one person's revisions aren't visible to all members of a work group (e.g., politically sensitive comments).

It was mentioned that the Adobe PDF document format allows selective modifications to a document.

Masinter: It is hard to see how a document can carry this amount of authorization.

Future plans of Word:

An Intranet Day presentation by Bill Gates demoed two features of an upcoming Word release.

Nielsen: The Amaya structured HTML browser and editor (W3C work performed at INRIA) does this already.

Masinter: The Shared Books system at Xerox might have done this too.

(General acknowledgment that these capabilities have existed in previous research systems.)

Some Word 8 features:

- automatic URL recognition (and conversion into a browsable link)
- common office hyperlink model across Office (both user-interface and programmability)
- web toolbar
- browser integration (the browser is opened using the DocObjects extensions to OLE, e.g. Explorer)
- document map (a frame-oriented table of contents style capability)

One problem they have been experiencing while working on Word is how to make a user interface which is good for both browsing and authoring simultaneously. Certain authoring features are undesirable during browsing, for example, the feature which marks misspelled words with a wavy red underline -- the user doesn't want to know this for sites they didn't author. One solution is to have a mode-based interface, with distinct browsing and authoring modes. However, Fein would prefer to have editing and browsing occur within the same mode.

Long: We encountered this problem in NaviPress as well. Our solution is: you are in browsing mode until you place your cursor in the window, then you switch to browsing/authoring mode.

Another participant mentioned that Netscape Gold has an authoring mode which is separate from their browsing mode, even popping up a separate authoring window.

Fein discussed the desirability of having the file format (e.g., HTML, Word native) and connection type (e.g. LAN, SLIP) be visible, but still the same operation to the user, for most cases. For example, HTML should be just one option for saving, rather than having a separate 'Save to URL/HTML' command.

Since the native Word document type is a richer format than HTML, there is some information loss going from Word to HTML. There are also some aspects of HTML which don't convert exactly to native Word format. This causes a problem for 'out-of-band' editing of HTML, that is, editing an HTML document that was created by converting a Word document to HTML, and then wanting to recover those changes back into Word. This led into the broader issue of whether the Web should support all word processing features.

Brown mentioned that some of their customers really cared about the format of generated HTML because they were feeding the generated HTML into other tools.

The issue of multi-purpose authoring was also raised: how best to have just one document type which is good for CD-ROM, for the Web, for printing, etc. Especially for printing, bandwidth becomes an issue. It is acceptable to retrieve a high-resolution image over a LAN to produce high-quality printed output, but this is probably not acceptable when working over the Internet, since it is not as fast. Are users willing to wait half a day to download the high-quality files to produce a high-quality printout?

There was some discussion about how best to handle writing a resource which contains many related resources, such as a web page and associated graphics.

Masinter: The Web mail group has been working on a way of agglomerating many files into one file, using the multipart/related MIME type. This allows the sending of a web page (many files together to form one page) via email.

There was some discussion about the need for a "GET for edit" method which is distinct from a "GET of source" or a "GET for browse" (the current behavior of HTTP). The need for this becomes clear when revision information is stored in a resource. In this case, a user performing an edit may not want the full source, because they do not want to incur the time delay of downloading the full revision history, but they may not want the browsable version either, due to server-side include processing.

Interoperability:

Existing tools do not all share the same editing model. For example, with FrontPage, it is possible to have a company name stored in one place. When this company name is modified, the change is then propagated to N leaf nodes. This is considered to be only one change to FrontPage, but it might be considered N separate changes to another tool.

Another issue is defining behaviors for less savvy tools, for example, what should revisions look like to a non-revision savvy browser?

Integrations with existing versioning systems -- this topic should be handled by the versioning working group.

In conclusion, Fein stated that word processing applications already have collaboration features, which are not now available in existing Web authoring tools. Microsoft is committed to a standards-based approach to collaborative authoring. In the long term Fein sees Word becoming the main editing engine, while FrontPage will concentrate on its facilities for site management, offering a higher web functionality.

Schulert: I see FrontPage being a more high-end, more web-focused tool than Word.

Whitehead: Can you tell us which HTTP method will be used in Word 8 for its "Save to URL" capability?

Fein: Sorry, I cannot talk about this.

Nielsen stated that the Word team should use the PUT method rather than the POST method. Schulert responded with a discussion about why the POST method was chosen by FrontPage group. The primary reason is that when they began development of FrontPage, POST was implemented and standard across servers, while PUT was not. Nielsen stated that the HTTP 1.1 PUT is now much more usable for writing content to an HTTP server than the HTTP 1.0 PUT.

Masinter asked whether the HTTP 1.1 version of PUT is adequate for putting versions with metadata, and also about the atomicity of PUTs? Long suspects that HTTP 1.1 still has some problems in this area.

Nielsen: Need to aim for December timeframe to get into HTTP 1.2, since there are many products being worked on, and HTTP 1.2 appears to be the last revision of HTTP.

Masinter: This group needs to produce some input for the HTTP Working Group by the end of the summer to have results put into HTTP 1.2.

Distributed Authoring - Nielsen

Nielsen began by reiterating the need to work quickly to get results into HTTP 1.2.

Nielsen would like to be able to create a document once and then have it seamlessly: (1) sent to a friend via email, (2) posted to a newsgroup, and (3) written to a web page, all using a common PUT/POST model. Some issues this raises are what transaction model is used? What quality of service do you want to have? Nielsen used the example of having icons on the desktop representing people, web servers, news servers, etc. with the capability of dragging a document onto one or more of these icons and then saying "submit". This causes the appropriate email post, newsgroup post, and web page write to occur simultaneously with appropriate error handling when things go wrong.

Nielsen also favors using the LINK mechanism within HTML to have rich relationship semantics among documents. These links could serve a number of purposes, for example, serializing a hyperweb of documents so they can be printed, or linking a table of contents to chapters of document. These relationships could also be used to create alternate hierarchies to the containment hierarchy offered by the directory structure of most existing web servers. This more general model can handle web servers implemented on top of a database as well as on top of a file system.

In Nielsen's view, during a PUT, the client may make suggestions about the destination URL of a resource, but is not in charge of the name space of the server. The server ultimately decides where it will place a resource in its name space. The server uses the link relationships to determine the location and hierarchical arrangement of resources within its namespace, rewriting HTML as necessary to preserve the relationships.

Some meeting participants were unaware of the LINK tag, and so there was a brief discussion describing it, and what it can be used for. This led into a discussion of the

desirability of having a standard for LINK REL tags (usage convention). It was mentioned that Murray Maloney had previously written a document on standard relationships, and that the time might be ripe to revisit this.

Hamilton: What about HyTime? It can project structure onto a document.

Frystyk: Same with Hyper-G.

Hamilton: There is a need for addition of structure that is not embedded into a document, like a linkbase system.

There was some discussion about the advantages of using standard link relationships. Concern was raised that a full consideration of this functionality would take longer than the two months remaining before the end of the summer deadline for impacting HTTP 1.2.

Masinter: The two month timeframe is only for changes to HTTP. However, there is the PEP (Protocol Extension Protocol) which can support changes beyond 1.2.

There was some discussion about whether there is a need for two phase commit transaction semantics for PUT operations in HTTP.

Wills: Should versioning be tied to a particular versioning system, like CVS (Concurrent-Versions System)?

Nielsen: Versioning should not be tied to a particular system. Authoring tools should use PEP to get the operations needed to perform versioning with a given system. How a particular versioning system performs locking and access control will be negotiated between the authoring client and HTTP server in some meta language.

Whitehead: Using PEP is not necessary, since it is possible to implement several different versioning styles using a common set of atomic operations.

Masinter: Perhaps the variability between versioning styles can be expressed in standard ways, such as by using a forms-based interface.

Seiwald: Whitehead says we can do a lot with some atomic operations, which allow for basic interoperability across versioning systems, so perhaps we should at least do this.

Nielsen: I don't want to end up with an enumeration of particular version control systems, but by using a negotiation model common methods will emerge.

Masinter: Registration and negotiation is the fallback when standardization is not possible. It is better than chaos, but standardization is better than negotiation.

Whitehead: There are also significant technical problems with negotiation. For example a lock simply prevents writing to a resource by all users except the lock holder. Describing a lock thus requires an access control meta language, the description and implementation of which is a difficult problem.

Nielsen: What goes into HTTP has to be independent of the versioning model used. (There was agreement on this point.)

Back to presentation:

Nielsen: Caching needs unique names for versions and variants so caches can deal with them uniquely.

Masinter: No, HTTP/1.1 doesn't require unique names. Could perhaps use a version header instead, which the cache could understand.

There was some discussion about the possible use of entity tags in a versioning scheme. Masinter: There should be some investigation of if-match and PUT usage. There doesn't really need to be that much support for versioning. There is a tradeoff in putting a new version header on the request versus putting version identifiers into the name space.

Nielsen mentioned that PUT does not support writing to resource byte ranges (i.e., sub parts of a resource).

Nielsen also mentioned that the Jigsaw server has an integration with CVS, and that the Amaya browser/editor will be 1.1 compliant as quickly as possible.

Nielsen also mentioned the need to make PUT reliable in practice, by which he means ensuring that PUT doesn't drop data on the floor during a transaction (lost data is bad).

There was some discussion about the applicability of SHTTP (Secure HTTP) for secure writes. According to Masinter, the SHTTP specification is in last call and will be approved very soon. Nielsen expressed the opinion that SHTTP is very difficult to implement.

Afternoon (1:35 PM)

AOLpress and AOLserver - Long

Long started by informing everyone that GNNpress and GNNserver are now known as AOLpress and AOLserver.

In AOLserver, the current version of resource is stored in the file system, while the meta-data, indices, etc. are stored in a database.

Long noted that people think of their web sites in terms of a hierarchy, and hence it is disorienting for the server to place a resource at a location other than the one specified by the user.

With regard to server-side includes and other dynamic content, AOL has no strong solution for getting to the actual source. Their current approach disallows editing a page with dynamic content. However, AOLpress users seem to be doing it anyway, although it is not clear what mechanism they use. His recommendation for how to get a document's source before server side includes processing is to use content negotiation when requesting the source, and asking for MIME type text/x-html-ssi. Unfortunately, this approach does not handle the case where the source is not HTML.

A discussion then ensued about pretty printing (canonicalizing) the HTML written by an authoring tool. There were some questions about how much canonicalizing an authoring tool should attempt. Long mentioned that AOL used to get grief from customers because their tool automatically cleans up its HTML output, but they no longer do, since the tool now additionally pretty-prints the HTML source. It appears that users perceive enough extra value in the pretty-printing of the generated HTML that they are willing to accept the cleanup.

Masinter: There are people who are claiming to do HTML authoring who want their mail client and mail servers to send around marked-up mail. They are using a kind of HTML that they make human readable through operations such as centering HTML source text between <CENTER> tags.

Whitehead: Is there an interoperability concern with regard to pretty printing?

Long: Probably not. I would be happy if we just made it so that everybody's client could do a PUT to everybody's server. That would be the most basic form of interoperability -- everything else is icing on the cake.

Nielsen: The W3C could put up a playground where everyone could test out the PUT interoperability of their authoring tools, so long as we don't get into the test suite business.

Long mentioned that the focus of his talk is not interoperability. He followed Dan Connolly's suggestion to go through your system and find out where you feel fuzzy and the least likely to be doing something in the same way as everyone else.

Continuing his presentation, Long described how moving pages brings up the issue of how to move relative links. Moving image files is easy, because the IMG tag indicates the image is considered part of the document, and should be moved. However, moving audio files is not as easy because they are included via a A HREF tag, which could refer to a close or distant resource. The best solution to this problem is to employ collections and move mini-webs. Unfortunately, many beginning users do not use the AOLpress mini-web functionality. Another solutions is to have a usage practice where relative URLs are employed for references internal to a collection and external URLs are used for references outside of it.

This led into a discussion on the semantics of "/x/y" style relative URLs
I.e., they're not "../foo" or "http://.."

Current versioning implementation slide:

Long reported on a prototype effort to add versioning capability to the AOLserver. In this prototype, reads employ a "Content-Version" header to retrieve a stated version of a resource. Writes provide a "Derived-From" header so the server knows how to check-in the version of the resource. The server prevents dirty writes. This implementation solves the "CREATE" problem (i.e., it does not need a separate CREATE method for the initial creation of a resource, because specifying a null Derived-From header directs the server to perform a resource creation). Since this prototype was completed, the header syntax has changed for HTTP 1.1. An incomplete report on this is available at URL:

<http://staff.navisoft.com/users/d/dave/w3c/wd-derived-960128.htm>

As an aside, Long mentioned that users really wanted to know if they were going to overwrite a document. This was accommodated by providing a dialog box with "OK" and "Cancel" options whenever a document was about to be overwritten. This required performing a HEAD operation on the resource before a PUT could take place.

At this point there was a discussion of entity tags (Etags, an opaque string which identifies a snapshot), and their potential use in versioning. Long believes entity tags are sufficient to avoid lost updates. Masinter recommended as an action item the review of entity tags to see if they are adequate for versioning. Nielsen stated that use of entity tags may have some interaction with caching that would be undesirable.

Long continued his presentation by describing locking within AOLserver and AOLpress. The AOLserver implements LOCK and UNLOCK methods. If a resource is currently locked, a "Locked-By" header is returned stating who currently has the lock. During an unlock, if you have write permission on a resource, you can unlock it. If you do not have write permissions, a message is displayed stating that the resource is currently locked, and by who. An OPTIONS method is used to detect server support for locking.

Access Control slide:

- users/groups (multiple membership)
- METHOD URL (specific/inherited)
- machine address (netmask)

It is possible to (allow, deny) a (user, group, netmask) for a particular URL.

There is a forms based user interface for access control functionality. There was a discussion of the value of this approach. The main benefit is that clients do not need to understand the content of the forms, they just merely need to be able to put them up.

Versions slide:

Resource revisions are time stamped and saved. Revisions are accessed through a prefixed URL, with a time stamp in it. Relative links and images are resolved as of that revision's time stamp.

PUT vs. POST slide:

Long expressed his position that PUT is a much better method for writing content than POST. PUT simplifies access control. PUT simplifies infrastructure such as caching proxies, gateways, etc., since it is ambiguous whether an arbitrary POST is transmitting an order for a pizza or writing a resource. There are also general-use servers which are starting to support the PUT method.

Namespace methods slide:

- DELETE (This is now supported in HTTP 1.1.)
- MKDIR (In AOLpress they wanted to support a "Save As..." dialog box, which includes a new directory option, since people think of their web space as a file system.)
- BROWSE (A method which looks at the contents of a directory.)

An example of the result of a BROWSE method is the return of an application/x-navidir MIME type.

```
application/x-navidir  
{MIME type} {name}  
...
```

Dave then displayed a slide on the issue of whether we should standardize Server API's. As rationale for why this might be desirable, Long mentioned that if, as an author, you need to do something to the server for your pages, it would be nice to use an existing API rather than talking to the site administrator for every change. ISAPI was mentioned as a possibility.

Variants slide:

- each variant has its own name (there is a need to standardize the naming scheme)
- does this apply to versioning as well?

- Assigning variants to resources similar to access control

PICS support slide:

- data-driven support for metadata systems
- Should raters provide well-known URLs to forms-driven interfaces for generators that produce rating-specific encodings?

Nielsen mentioned that the W3C has a tool where you can plug-in a PICS rating system, and it pops up the correct buttons, sliders, etc., and allows you to set the rating for a page.

Miscellany slide:

- cross-server searching (WAIS, MARC, ...)
- cross-vendor auto-linking? (Search the web, automatically create links, or give choices of what you might want to link to).
- better support for link checking.

A question was raised about redirecting URLs, and how to tell if a resource has been permanently moved or only temporarily moved.

Brown: It would be useful to specify what is the guaranteed URL.

Masinter: You can register your URLs with the OCLC Persistent URL Service at URL:

<http://www.purl.org/>

Nielsen: You can abstract the Web space from the file space by having the server remap the resources that it owns.

FrontPage - Schulert

Release 1.1 was shipped this Spring, and release 2.0 will be shipped sometime this Fall. Vermeer was bought in January by Microsoft.

FrontPage was a client-server web authoring tool from the beginning. Ideally, FrontPage wants to be server independent -- they do not want to be in the server business, and would prefer to use HTTP as the base upon which they build FrontPage.

When they first started development of FrontPage, using the POST method was their only choice, since it was widely available on existing servers. They implemented an RPC-like mechanism on top of POST. There are somewhere between 20 and 30 entry points into FrontPage using this RPC-like mechanism. FrontPage has three server extensions executables, one for each of the three levels of access control available.

During design, FrontPage was aimed more towards the mainstream user, rather than the power user. The design was approached as a standard client-server development, where functionality was placed on whichever side of the wire it most made sense.

The FrontPage server extensions include much functionality, such as the ability to return a link map for the site, enhanced semantics for some operations, and the ability to set access control from the client. It does not include all features of SSA client control. There are capabilities for link repair, such as being able to inform FrontPage that a given URL has

changed, and having FrontPage fix it in all of the pages on the server. There is a bulk upload (multi-resource at a time) capability.

FrontPage features webbots, which are objects that can be dropped into HTML and which are active when a page is uploaded to the server. An example webbot is the "day last changed" bot, which automatically inserts the date the web page was last changed into the source HTML for the page. Other webbots include a "substitution" bot, which can, for example, replace a company name in all web pages on a server, and a "table of contents" bot.

Schulert stated that the FrontPage group would like to get out of the business of maintaining the CGI scripts and server extensions.

Whitehead: How tightly are the webbots tied to using POST?

Schulert: We just need a way of invoking the webbot behavior.

Masinter: If you want to get out of maintaining those three CGIs, could you make FrontPage work with PUT?

Schulert: We can see a path out of it, using server-side Java or Visual Basic.

Schulert stated that you can post a web up to a server using FTP. You can post just changes that way, too. This solves the problem of directly editing the server when you don't want to make direct changes and a versioning system is not available. Hence we could use PUT in a very basic way, like using FTP, to have an interoperable write capability with other servers.

FrontPage currently has a bundled search engine, but Schulert does not view FrontPage as being in that business either -- the search engine is just enough to provide basic functionality. Ideally, Schulert would like version control, and search engines to be pluggable on the server.

In FrontPage 1.1, there is conflict detection and collision prevention, but no resolution help.

There was some discussion surrounding the problems of what you author is not what you are getting -- there can be things that the author knows that the server doesn't get. For example, if I delete a slide from a presentation, or a message in a discussion, the adjacent next and previous links are munged, and the server may know nothing about remedying that. Schulert sees there being an HTTP-level problem, in the long term, in this change of abstraction and also having conflicting server-side extensions. How do people's extensions cooperate with each other without knowing about each other?

Schulert stated that there is a core set of functionalities beyond PUT.

Suggestions:

- PUT is a great first start
 - Can post a web to a server using FTP using FrontPage 1.1
- PUT isn't good enough. Neither is GET.
- PUT would be the equivalent of their FTP posting capability.
 - FrontPage 1.1 allows operation in a mode where FrontPage does not control access control.

Goal of Working Group - Whitehead

Whitehead next led a discussion about the goals and membership of the working group.

Whitehead displayed a slide which stated that the goal of the working group should be to make distributed authoring as pervasive as browsing is today.

Brown: I don't know about pervasive.

Long: I keep getting E-mail from customers who are using Netscape and notice a typo or spelling mistake and want to fix it. That's the world we want to address.

Masinter: We could be modest and not change the world, and simply have interoperability among our tools.

Fein: But we want to have interoperability for things we would like to do, not just what we are trying to do.

Masinter: Yes, we should look about two years ahead.

Nielsen: We should have something at level 0 very quickly -- I can't see farther than six months out.

Whitehead then recommended that the group adopt as its goal/objective/aspiration: to ensure that distributed web content authoring tools are broadly interoperable.

Masinter: We should change ensure to enable, so it would read: enable distributed web content authoring tools to be *broadly interoperable*.

Seiwald: We ought to keep that goal in mind, as a mission statement.

There was a suggestion to add a statement about standardizing features that exist today.

Nielsen asked that interoperability be limited to the HTTP framework.

There was a discussion about what was meant by interoperability, and what level of interoperability should be strived for by the working group. From this discussion, it was clear that there are different kinds and scopes of interoperability, and that working towards broad interoperability was a reasonable statement of the group's goal.

The working group adopted as its goal:

Enable distributed web content authoring tools to be broadly interoperable.

Sponsorship of Working Group by World Wide Web Consortium

Whitehead next led a discussion about whether this working group should seek sponsorship by the W3C. Nielsen mentioned that the W3C has a web page on the process used to create a W3C working group, at URL:

<http://www.w3.org/pub/WWW/Consortium/Process/WorkingGroup.html>

Nielsen mentioned that being a W3C working group implies that you have to honor the W3C agreements, including the meeting procedures, and it has to work on a focused output.

Seiwald: What is the difference between being 14 random people or being affiliated with the Internet Engineering Task Force (IETF)?

Masinter: If you are a working group, there are legal safeguards. For the IETF you have to have a chair, an editor, and a proposal and a draft. They don't require an initial meeting, and you are subject to the approval of the IETF Steering Committee for being appropriate to the IETF.

Nielsen: Becoming a W3C working group makes sense because it is within the domain of the area of interest of W3C.

Seiwald: Does the result go into the IETF?

Masinter: You can, if you need to make an Internet standard. There must be an open public review at the IETF, so that is another gate. This effort shouldn't be uncoordinated from the HTTP Working Group, however, it is not just HTTP that is of concern here.

Whitehead favors W3C sponsorship since it is the natural focus on Web-related issues. Hopes to not have quite the formality of IETF.

Fein: What does this obligate us to on behalf of our employers? What other conditions are there?

Masinter: If you work for a company that is a member of W3C, then your company has already aligned with the W3C agreements.

Hamilton: I just want to know what the ground rules are, rather than finding out retroactively.

Whitehead stated that he will investigate the intellectual property rights issues of non-W3C members participating in a W3C working group.

Meeting Goals, Criteria for Completion - Whitehead

Whitehead next put up a slide listing desirable meeting goals, and started a discussion on what would constitute criteria for completion of the working group's activity.

Masinter mentioned that he likes the idea of a having a demonstration of interoperability. For example, multiple authoring tools working against multiple servers and one document being edited by multiple authoring tools. It would be desirable that there be round-trip preservation of content features. For this demonstration of interoperability, the authoring tools should of the same level of quality that people already have.

Whitehead and Burns agreed that this is certainly ambitious, and it doesn't require versioning, or other advanced capabilities.

Burns: Having this demo actually run seems like a proper exit condition

Masinter: The IETF condition for advancing to a draft standard is that there be multiple interoperable implementations

Whitehead mentioned that a second demo is that N authoring tools do simultaneous editing of the same resource on one server, which would test lost update capabilities, collision handling, etc.

Brown: It appears there are several cases:

1. A single client used against multiple servers.
2. Multiple clients used against a single server.
3. Multiple clients used against multiple servers.

Whitehead: We will need to refine these cases to make them more concrete.

Long: The deliverables are specifications by which each of these demonstrations can claim conformance.

Whitehead: Let's look at key interoperability issues before we specify deliverables.

Key Interoperability Issues (Nielsen's List)

Nielsen wrote a list of what he considered to be the key interoperability issues on the whiteboard in the conference room. He openly admitted that there was significant bias in his choice of issues.

- Semantic Links
- Make sure that PUT, DELETE work
- What you "GET" Is not what you author - GET for edit versus GET for browse?
- Ability to Edit Dynamic Contents
- Content-Version, Derived-From, ETAG
- Security (digest, content-MD5, SHTTP)
- Access Control Management
- Link Management on Server
- When to Publish (Publish header)
- Do we need transactions
- Advanced Editing (Multiple users, roles, views)
- Adding revision and change information into HTML

The HTML item was added during group discussion. These interoperability issues then became the focus of discussion.

Schulert proposed that the GET-PUT working bullet is the following example: A user is browsing at a web page and sees a misspelling and wants, with simple steps, to edit the page and put the corrected page back out on the web.

Nielsen offered a scenario where there are a set of interrelated resources, and putting them back to a server atomically is desired.

Masinter: There are tasks below those - let someone author a new page, let someone do site maintenance.

Masinter then recommended that we go around the room and see what work people are willing to do as part of the working group.

Whitehead stated that he will write meeting notes for this meeting, but is not taking ownership of individual technical issues due to the number of issues, the amount of work required to coordinate the group, and his desire to concentrate on versioning issues.

Masinter is willing to help coordinate the activity of the working group, and contribute as an active member of the mailing list. He does not want to edit anything new.

Seiwald: I have more interest in the HTTP part than the HTML stuff and would be interested in writing up proposed HTTP changes. Willing to author proposals as needed.

Fein: There is more interest in HTML issues than HTTP issues in my case. I'm not as interested in protocols or back ends as I am in document content itself. I could take the Word requirements for HTML and write them down.

Masinter requested that Ron sort them out as content versus protocol etc. A good distinction between what is expected behavior for understanding, for not understanding, and for sort-of understanding a tag would be helpful.

Nielsen: That goes for all existing distributed authoring tools.

Dawson and Nielsen: Fein, Schulert, and Long all have lists of what they want to do in authoring tools - Word, FrontPage, and AOL. They could take on a task to sort out requirements.

Hamilton: I propose that Fein, Schulert, and Long (Word, FrontPage, AOL) use Fein's model (employed on his slides) for listing functional requirements and features and possible solutions.

Long and Burns are going to work on assembling something that is wordsmithed on functional requirements and scenarios. The tasks / scenarios that provide the cover for the features list.

Nielsen mentioned that he has three scenarios at three different levels in his slides.

Dawson volunteered to edit the scenarios document.

Seiwald invited everyone to submit two scenarios, not including the three we have.

Masinter: The task scenario document could be long. There is nothing wrong with that. We should also invite people to contribute. There may be substantial contributions from others who are not here.

Nielsen: I think coming up with a requirements document is nice. However, there are some things that need to be fixed and we could also start working now on the technical issues.

Masinter: We don't have to wait for the scenario document to be complete to start work on the technical issues. However, we will need the scenarios document to tell other people why we are doing this and how we will know when we are done.

Looked at Get for Edit versus Get for Browse. It's important to lay out what the alternatives are so you can say you discarded them.

Nielsen gave his suggestions for scenarios which should belong in the scenarios document:

1. One person changing a misspelling in a document they found while browsing.
2. Checking-in a group of resources which are related.
3. Deleting an object from a web.
4. Two people editing changes to the same resource.

Specified Deliverables

The group came to agreement that the following set of activities and deliverables should be produced by the working group:

1. Task-oriented list of scenarios which interoperable distributed authoring tools will be able to perform.
 - Keith Dawson, editor.
 - Input of other distributed authoring vendors, notably Netscape, will be solicited.
 - Put a call on the discussion list request input for this activity.
 - Nielsen will give us his list of requirements.

2. Collate lists of "key functionality" among AOLpress/AOLserver, FrontPage, Word, as well as other distributed authoring tools, such as Netscape
 - Dave Long, editor

Looking at early September for next meeting, for at least the groups working on the deliverables above.

*** Meeting Adjourned ***

Slides: Jim Whitehead (Agenda, Background, Discussion)

Working Group on Distributed Authoring on the World Wide Web

AOL Productions
San Mateo, California
July 10, 1996

Agenda

Morning (9AM)

- Introductions
- Discussion of the direction and purpose of the working group
- Overview of efforts to date on distributed authoring
- Review of current distributed web authoring systems and the mechanisms they employ:
 - + Henrik Frystyk on Amaya, PUT model
 - + Dave Long on GNNpress/GNNserver
 - + Andy Schulert on FrontPage
 - + Ron Fein on collaborative web authoring, MS Word

Agenda

Afternoon (1:30PM)

- Identification of key areas in which existing systems are unable to interoperate
- Discussion of which key areas can be profitably addressed by this working group, and a rough prioritization of the areas
- Solicitation of volunteers for coordinating work on improving interoperability in key areas
- Development of criteria for evaluating the success of key area work groups
- Establishing timelines for action

Discussion of Direction and Purpose of Working Groups

- The discussion that follows gives the best current description of the purpose for:
 - ✦ Distributed Authoring Working Group
 - ✦ Versioning and Configuration Management Working Group
- The purpose and sponsorship of these groups is open for discussion

Distributed Authoring Group

Goal: to make distributed authoring as pervasive as browsing is today.

How?

- By specifying:
 - ✦ Preliminary modifications to existing specifications (e.g., HTTP)
 - ✦ Usage conventions (e.g., URL name space)
- Ensure that distributed web content authoring tools are *broadly interoperable*

Distributed Authoring Group

- Web Page:
<http://www.ics.uci.edu/~ejw/authoring/>
- Mailing List:
w3c-dist-auth@w3.org
- Sponsorship:
 - ✦ Currently an informal group working in cooperation with W3C
 - ✦ W3C would like to transition this into an official W3C Working Group

Distributed Authoring Issues

- A non-exhaustive list of issues which need to be addressed to achieve interoperable distributed authoring tools:
 - ✦ Access control
 - ✦ How to perform a distributed write
 - ✦ Access to “raw” HTML before SSI’s
 - ✦ Versioning and Configuration Management
 - ✦ ...

Versioning and Configuration Management Working Group

Goal: develop a preliminary specification for how to provide versioning and configuration management of content served by an HTTP server.

- Accomplished by the same means as distributed authoring group: extension of existing specifications, usage conventions

Versioning and Configuration Management Working Group

- Web Page:
<http://www.ics.uci.edu/~ejw/versioning/>
 - ✦ Collects all known approaches to the WWW versioning problem
 - ✦ Provides background material from hypertext versioning and SCM literature
 - ✦ Historical documents
- Mailing List:
www-vers-wg@ics.uci.edu

Versioning and Configuration Management Working Group

- The working group is not officially sponsored by either the IETF or the W3C
- However, the group does have as participants, members of the following:
 - ✦ Major document management corporations (Alpharel, Documentum, Filenet, Network Imaging, Vantech, Xerox)
 - ✦ Major software config. management vendors (Atria, Continuous, MKS, P3)
 - ✦ All active hypertext versioning researchers
 - ✦ IETF HTTP WG (Larry Masinter), W3C (Dan Connolly), Netscape, Microsoft, Apache (Fielding, Thau), NTT Lab, many others...

Issues:

- Does the stated purpose of the distributed authoring group correctly encompass your expectations of what this group should accomplish?
 - ✦ Broaden scope? Limit scope?
- Should we seek W3C sponsorship?
 - ✦ Recommendation: Yes

Distributed Authoring History

- Source: Received view from W3C (Dan Connolly)
- 1989/90: Nexus web authoring tool
 - ✦ most of the world sees line-mode browser: not impressed
- 1992: Mosaic 2.4 reaches critical mass.
 - ✦ "Publish/Browse" becomes the dominant model.

Distributed Authoring History (cont'd)

- 1992: PUT disappears from HTTP specification, due to lack of support.
- 1994: NaviPress/NaviServer developed
- 1994(?): Vermeer FrontPage developed
- 1994/1995?: W3C Line Mode Browser
- Dec '94: WWW4 breakout session on distributed authoring tools
 - ✦ <http://www.w3.org/pub/WWW/Collaboration/9512www4/auth-tools>

WWW4 Breakout Session (Dec. 1994)

- Focus of session: Interoperability among distributed (i.e. client/server) web authoring tools
- Identified interoperability issues:
 - ✦ Common Access Control Model
 - ✦ "Lost Update" problem
 - ✦ New HTTP Methods: BROWSE, MKDIR
 - ✦ Editing Variants
 - ✦ Access to "raw" HTML before SSI processing
 - ✦ Strong authentication (e.g. MD5)
 - ✦ Place HTML DTD (w/ URL) in HTML source

Distributed Authoring History (cont'd)

- 1995/96: Rohit Khare at W3C researches a leasing and locking mechanism
 - ✦ <http://xent.w3.org/~khare/6.852%20Project.html/index.html>
- March 1996: Dan Connolly puts out a call for volunteers to coordinate distributed authoring activity
 - ✦ Jim Whitehead, U.C. Irvine, volunteers
- June 1996: IETF HTTP working group completes internet draft of HTTP/1.1
 - ✦ PUT method is part of standard

Distributed Authoring History (cont'd)

- July 1996: IETF HTTP working group completes internet draft on digest authentication
- July 1996: Meeting on distributed authoring on the WWW at AOL Productions, San Mateo, CA.

Key Interoperability Issues

- Access control
 - ✦ needed to support remote visibility management
 - ✦ versioning styles with locking need it
- Method of performing a distributed write
 - ✦ HTTP PUT vs. HTTP POST vs. FTP?

Key Interoperability Issues (cont'd)

- Versioning support
 - ✦ Support for parallel development (avoid lost updates)
 - ✦ Access to old revisions of a resource
 - ✦ Browse a versioned collection of resources
 - ✦ Managing collections of versioned resources
- Access to “raw” HTML
 - ✦ How to fetch HTML without server side include processing

Key Interoperability Issues (cont'd)

- Editing variants
 - ✦ How to retrieve/write them? Which URL?
- Directory Creation
- Determining which HTML DTD was used to create/parse an HTML source
- Other issues identified today...

Meeting Goals

- Determine the overall goal of the group
 - ✦ Membership criteria
- Establish criteria for completeness of the group's work
- Determine deliverables with target dates
- Expected amount of W3C resources which the WG will take to run
- Commitments from non-W3C staff upon which the running of the WG is predicated.

Membership

- The process should be open with wide participation from all parties who have expertise or insight into the issues.
- Representation from W3C member companies, since they have expertise in this area
- People from outside the W3C who have expertise in the area should be encouraged to participate as well.

Criteria for Completeness

- What constitute the criteria for the completion of this group's activity?
 - + Connolly: Done when there's a shared understanding of distributed authoring mechanisms in the development community and the market.
 - For example, the top 3 to 5 editing clients and servers all interoperate according to published specs.

Collaborative authoring in Microsoft Word

Ron Fein
Microsoft Corporation
July 10, 1996

Overview

- **Collaborative authoring features in existing word processors**
- Word's future plans
- Directions for standards-based development

Current document collaboration features

- Revision marking
- Annotations
- Merge documents
- Compare documents
- Access control

Revision marking

- User problem
 - Track changes per-edit
 - Protect document against unauthorized changes
- Solution
 - Provisional edits stamped with user/time
 - Storage independent of display/print
 - Fully deferrable review process
- Web considerations
 - Edit records increase file size

Revision marking example

- Before

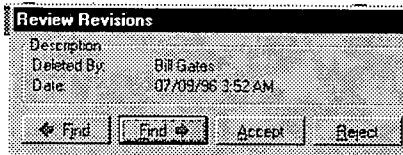
We will never have an
Internet division.
-Bill Gates

- After

| We will soon~~never~~ have
an Internet division.
-Bill Gates

Revision marking example

- Review




| We will soon~~never~~ have
an Internet division.
-Bill Gates

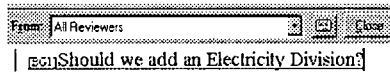



Annotations

- User problem
 - Add/view comments to specific ranges of text
 - Unlike revisions, these are not proposed changes to the document
- Solution
 - Separate document stream anchored to ranges of main text
- Web considerations
 - Should browsers display these?

Annotations example

Annotation

We will soon never have
an Internet division. 
-Bill Gates

 From: All Reviewers   
| Should we add an Electricity Division?

Merge documents

- User problem
 - Allow reviewers to integrate multiple sets of revisions to same document
- Solution
 - Add edit records from all sources to main document
- Web considerations
 - Requires revision marking
 - Until review, document in half-finished state

Compare documents

- User problem
 - `diff` for arbitrary documents
- Solution
 - Low-level compare
 - Heuristics for ambiguous differences
- Web considerations
 - Likely scenario unless there is universal revision scheme

Access control

- User problem
 - Provide granular permissions (read, edit, turn off revisions, print, etc.) to different users
- Solution
 - Username-based permissions list
- Web considerations
 - Security and server support

Word 8 and beyond

- Collaborative authoring features in existing word processors
- **Word's future plans**
- Directions for standards-based development

Word 8: the Intranet release

- “Word automates the writing process of common documents for individuals and workgroups.”
- Features include
 - Automatic URL recognition
 - Common Office hyperlink model
 - Web toolbar
 - Browser integration
 - Document Map

Beyond

- Are you browsing or authoring?
- File format and connection type should be totally irrelevant to the user
- Graceful degradation into HTML and into print

What we need to do

- Collaborative authoring features in existing word processors
- Word's future plans
- **Directions for standards-based development**

Feature list to support

- How do collaborative authoring features differ w/r/t traditional documents vs. Web?
- LAN/file system vs. Internet

What goes where?

- Document vs. server for history/control information
- File size/performance issues
- Extending HTML to represent authoring information
- Security

Interoperability

- Edit models
- Defining behaviors for less-savvy applications
- Integration with existing versioning systems

Key points

- Word processing applications already have rich collaborative feature sets
- These features can be ported to Web standards with varying ease
- Microsoft Word is committed to a standards-based approach to collaborative authoring

Wrap up

- How to reach me:
 - ronfe@microsoft.com
 - Ron Fein, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052
 - 206-703-1716
- Questions?



Distributed Authoring

Henrik Frystyk Nielsen
Web Consortium / MIT

W3C/AOL

07/10/96

1



Table of Contents

- User Perspective - Three Scenarios
- Server Side Perspective
- Authoring and HTTP
- Distributed Authoring at W3C
- Future Planning and Time Lines

W3C/AOL

07/10/96

2

■ *User Perspective - three scenarios*

- One User - many Protocols
 - The User builds a **Web** of recipients using drag and drop
 - Transaction starts when user hits **Submit**
 - The application sends the message using *NNTP*, *SMTP*, and *HTTP*
 - Context defined transaction model to handle results

W3C/AOL

07/10/96

3

■ *Three Scenarios (2)*

- One User - many Documents
 - Define groups of HTML documents based on **<LINK>** tag
 - Link semantics supports printing, automated Web traversals and much more
 - The client doesn't control the URLs but *does* control the link relationships
 - Do not rely on URL hierarchy!

W3C/AOL

07/10/96

4

Three Scenarios (3)

- Many Users - many Documents
 - What is the granularity of authoring entities?
 - » A single document
 - » subparts of a document (paragraph)
 - Advanced User Role Management
 - Annotations
 - Notifications

W3C/AOL

07/10/96

5

Server Side Perspective

- Version Control Provided by X systems:
 - Existing document management systems (CVS, SCCS)
 - Data base modules
 - Ad hoc plug-ins
- Link Management
 - Storing and updating link semantics
- Access Control Management
 - Users, groups, ACLs

W3C/AOL

07/10/96

6

■ **Authoring and HTTP**

- Caching
 - Fine if each version and variant has its own name
- Versioning
 - Fine if opaque to clients and Proxies
 - *Content-Version* is enough!
- Security
 - Must control headers as well
 - *Content-MD5* is not enough!

W3C/AOL

07/10/96

7

■ **Authoring and HTTP (2)**

- PUT is not fully specified
 - Does not define use of Bytes Range
- What about HTTP Extensions?
 - Few extensions will cover all CMSs:
 - » The result of using locks and versioning is system dependent
 - » Client must know the consequences
 - Focus on the generic case in HTTP
 - PEP extension for each CMS can provide higher level functionality

W3C/AOL

07/10/96

8

Distributed Authoring at W3C

- Jigsaw

- Java based extensible server
- Supports HTTP/1.0 PUT and CVS
- Form based Authentication with users and Groups
- Basic and Digest Authentication

- Amaya

- C based structured HTML browser/editor
- Uses libwww for handling HTTP
- Supports HTTP/1.0 PUT

W3C/AOL

07/10/96

9

Future Planning and Time Lines

- HTTP/1.2 is due by December

- Fully specified PUT
- Version
- Content-MD5 equivalent for headers

- Open Issues

- Link Relationships
- Access control management
- Publishing control (when to *enable* the change)

- What resources do we have?

- Who can provide specs and code?

W3C/AOL

07/10/96

10

Slides: Dave Long (AOLpress/AOLserver)

AOLpress/AOLserver

Dave Long
America Online, Santa Barbara
dave@sb.aol.com

AOLserver "webstore"

- current version of resource in filesystem
- meta-data, indices, etc. in database
- DBMS not required; "stackable" with other servers

HTML source

- With server-side includes and other dynamic content, how does one request the "original" page?
(Accept: text/x-html-ssi ?)
- How much canonicalizing should an authoring tool attempt?
(more palatable when pretty-printed, line-break conventions preserved)

Rewriting relative links...

- duck the issue - do nothing
- duck the issue - use collections
- user-specified (state space?)
- template specified

Rewriting relative links...

- user-specified: by type
 - convert { none, relative links, all }
 - convert relative to absolute
- template-specified: for <APPLET> specifies which <PARAM>s can be rewritten
(as above)

Lost Update

- do nothing (ostriching)
- detect lost updates (versioning)
- prevent lost updates\tab (locking)
- fairness issues (leasing)

Lost Update - versioning

- reads include "Content-Version:" headers
- writes provide "Derived-From:" headers
- server prevents dirty writes
- solves the "CREATE" problem
- header syntax has changed for HTTP/1.1

Lost Update - locking

- LOCK/UNLOCK methods
- "Locked-By:" header
- use OPTIONS to detect server support
- runs on top of Content-Version scheme
- fair UNLOCK? collaborative groups hopefully share trust!

Access Control

- users / groups (multiple membership)
- METHOD URL (specific/inherited)
- machine address (netmask)

Access Control List

- METHOD+aliases URL (+inheritance)
- allow/deny (user0 user1 ... userN)
- allow/deny (group0 group1 ... groupN)
- allow + deny (netmask0 netmask1 ... netmaskN)

Versions

- resource revisions are timestamped and saved
- accessed through prefixed URL, therefore
- relative links, images, resolved as of that revision's timestamp
- absolute links? version != snapshot?

PUT vs. POST to publish

- PUT method simplifies access control
- PUT method simplifies infrastructure
caching proxies, gateways, etc...
- general-use servers starting to support PUT

namespace methods

- DELETE
- MKDIR
- BROWSE

BROWSE sample

```
HTTP/1.0 200 OK
MIME-Version: 1.0
Date: Tuesday, 09-Jul-96 23:35:44 GMT
Server: NaviServer/2.0 AOLserver/2.033
Content-Type: application/x-navibrowse
Content-Length: 289
application/x-navidir .
text/html .odo.index.htm
application/x-navidir download
text/html download.html
text/html feedback.html
application/x-navidoc images
text/html index.html
application/x-navidoc press
text/html search.html
application/x-navidoc server
application/x-navimap tools.nvm
```

URL mapping

magic URLs used for:

<http://server/NS/operation/path...> (or form data)

- page permissions
- page versions
- administer server
- PrimeHost Advanced Control Panel
- use methods instead?

PrimeHost Control Panel

linked to server administration page, provides

- log analysis, log rolling
- MIME type configuration
- FTP reporting, configuration
- email configuration: aliases/mailing lists

transparent extension!

Server API's

- should we standardize?
- can "stack" servers for some functionality
- ISAPI is a possibility

Editing variants

- editable variants have their own URLs
- does this apply to versioning, as well?
- assigning variants to resources similar to access control

PICS, etc...

- data-driven support for metadata systems (such as PICS rating)
raters provide form-driven well-known URLs to produce their encoding?
- old NaviPress "Describe" functionality
(forms-based association of metadata with resource)

miscellanea

- cross-server searching (WAIS, MARC, ...)
- cross-vendor AutoLinking?

better support for link checking (spec'd, just needs to be implemented?)