World Wide Web Distributed Authoring and Versioning (WebDAV): An Introduction

E. James Whitehead, Jr.
University of California, Irvine

■ Today, the typical use of the World Wide Web is to browse information in a largely read-only manner. But this was not the original idea—as early as 1990, a prototype Web editor and browser was operational on the Next platform, demonstrating how Web content could be read and written. Unfortunately, most of the world never saw this editor/browser, but instead developed their view of the Web from the widely distributed text-based line mode browser. When NCSA Mosaic was developed, it improved the line mode browser by adding a graphical user interface and inline images, but had no provision for editing. As Mosaic 2.4 reached critical mass in 1993–4, "publish/browse" became the dominant model for the Web. But the original view of the Web as a readable and writable collaborative medium was not lost.

In 1995, two browser/editor products were released: NaviPress by NaviSoft and FrontPage by Vermeer. These products began developing a market for authoring tools that allow a user to edit HyperText Markup Language (HTML) pages remotely [Raggett 1997], taking advantage of the ability to work at a distance over the Internet. In early 1996, NaviSoft and Vermeer were purchased by America Online and Microsoft, respectively, presaging major corporate interest in Web distributed authoring technology. In 1995–96, Netscape released Navigator Gold, a Web browser/editor tool, able to publish pages to a remote Web server. 1996–7 also saw the release of Web-integrated word processors, with Microsoft Word 97, Lotus WordPro 97, and Corel WordPerfect 7, all with HTML editing and remote publishing capacities.

In this setting, an ad hoc collection of people interested in remote authoring (now known as the WebDAV working group) met at the WWW4 conference in December 1995, and then at America Online in June 1996. Comprised of developers working on remote authoring tools, and people generally interested in extending the Web for authoring, this group identified key issues in writing these authoring tools, and also found a pressing need to develop standard extensions to the HyperText Transfer Protocol (HTTP) [Fielding et al. 1997] for the following capabilities:

—Metadata, to create, remove, and query information about Web pages, such as its author, creation date, etc., also to link pages of any media type to related pages.

—Name space management, to copy and move Web pages, and to receive a listing of pages at a particular hierarchy level (like a directory listing in a file system).

—Overwrite prevention, to keep more than one person from working on a document at the same time. This prevents the "lost update problem" in which modifications are lost as first one author and then another writes changes without merging the other author's changes.

—Version management, to store important revisions of a document for later retrieval. Version management can also support collaboration by allowing two or more authors to work on the same document in parallel tracks.

From its inception, the WebDAV working group has worked steadily to produce an interoperability specification that defines HTTP methods and their semantics for the above capabilities. Work in this direction has focused on three documents: a scenarios document [Lassila 1996], which gives a series of short descriptions of how distributed authoring and versioning functionality can be used, typically from an end-user perspective; a requirements document [Slein et al. 1997], which describes the high-level functional requirements for distributed authoring and versioning, including rationale; and a protocol specification [Goland et al. 1997], which describes new HTTP methods, headers, request bodies, and response bodies, to implement the distributed authoring and versioning requirements. WebDAV is a working group within the Application Area of the Internet Engineering Task Force (IETF) and works cooperatively with the World Wide Web Consortium, which provides technical assistance and help in contacting interested people within the Web community.

In the remainder of this article I'll present the rationale for developing a standard for Web distributed authoring and versioning, followed by a detailed overview of the proposed capabilities. The term "resource" is used throughout this article, which is the proper Web term for any piece of information such as a Web page, a document, a bitmap image, or a computational object stored on a Web server, and whose location is described by a Uniform Resource Locator (URL) [Berners-Lee et al. 1994].

Rationale for a WebDAV Standard

While open standards are currently very popular, in part due to the success of the Internet, it is worthwhile to examine the reasons for developing an interoperability standard for Web distributed authoring and versioning. There are two primary arguments in favor of development:

(1) An interoperability problem due to nonstandard HTTP extensions employed by commercial HTML editing tools exists, which will only get worse if an interoperability standard is not developed.

(2) Intranet-enabled applications want to provide Web-based collaboration support, but currently have no standard way to do so.

EMERGING INTEROPERABILITY PROBLEM

Today, the HTTP (version 1.1) protocol contains functionality which enables the editing of Web content at a remote location, without direct access to the storage media via an operating system. This capability is exploited by several existing HTML distributed authoring tools and by a growing number of mainstream applications (e.g., word processors), allowing users to write (publish) their work to an HTTP server. To date, experience from the HTML authoring tools has shown they are unable to meet users' needs using the facilities of the HTTP protocol. The consequence is either postponed introduction of distributed authoring capability or the addition of nonstandard extensions to the HTTP protocol. These extensions, developed in isolation, are not interoperable.

Distributed Web-content authoring tools that directly write content to an HTTP server currently exist—for example, Microsoft FrontPage, America Online AOLpress, and Navigator Gold and Communicator from Netscape. While these exciting tools demonstrate the enormous potential of a writable Web, they also embody the interoperability pitfalls that lie ahead. While all three tools can now use the HTTP "PUT" operation to write their content to a remote server, all three products use differing, custom HTTP extensions for non-standard functionality. For example, both AOLpress and FrontPage use different HTTP extensions to find out which documents are located at a remote site, and both the FrontPage server extensions and AOLserver (the companion server to AOLpress) handle access control differently. One cannot currently use FrontPage with AOLserver, or use AOLpress with the FrontPage server extensions. Furthermore, Novell and Lotus recently announced plans to offer Web servers with authoring support later this year; foreshadowing more serious interoperability problems still to come.

COLLABORATION INFRASTRUCTURE

Along with the incredible growth of organization-specific Webs, or intranets, a new class of tool is emerging: the Web-enabled tool. Just as distributed Web content authoring tools currently write to HTTP servers, a large class of more traditional mainstream applications can now read and write information from HTTP servers. For example, Word 97 and WordPro 97 can import and save HTML documents via HTTP. Web-enabled spreadsheets are also available: 1-2-3 97 and Excel 97 are both able to import data via HTTP and to export spreadsheets in HTML via HTTP. Though limited to simple HTML import and export capabilities right now, these tools are poised to take advantage of the WebDAV specification once it is complete, and provide network-based collaborative editing features. The collaboration infrastructure provided by the WebDAV specification lays the necessary foundation for provision of collaborative features to users, but the lack of such a standard prevents introduction of these tools today.

Functionality Overview

The WebDAV group has proposed a set of features that can be used in a wide variety of settings by applications that support collaborative work on remotely authored and versioned documents. These features can be partitioned into four groups: metadata, name space management, overwrite protection, and versioning. A detailed overview of these capabilities along with a snapshot of current plans for how these features should be specified are presented in the sections below. Many of the approaches described are still being actively considered by the WebDAV group, and hence should be considered a glimpse into work in progress, rather than a definitive statement. (The WebDAV functional requirements appear in a companion article, "Requirements for Distributed Authoring and Versioning on the World Wide Web," also in this issue.)

METADATA SUPPORT

Data on the Web has many pieces of associated information, such as the title, subject, creator, publisher, length, and creation date. Information about information (metadata) can be used to search for Web resources, enforce copyrights, or provide bibliographic information. Metadata is particularly useful in searching for Web resources, due to the inadequacies of existing index-based Web search engines which often return a large number of undesired results. By focusing a search on the value of a particular metadata field (e.g., the author), metadata can be used to reduce the number of undesired query results.

Development of a useful set of metadata is extremely important—one schema, or set of metadata, developed to assist Web searching is the Dublin Core [Weibel 1997]. Since other groups have focused on developing metadata sets, the WebDAV group decided to develop facilities for creating, modifying, deleting, and querying metadata. These facilities allow the manipulation of metadata from multiple schemas, hence the scheme may vary with the domain being used. For example, even though the Dublin Core is appropriate in the general Web context, it may not be ideal in other settings, such as the legal community. By being schema-neutral, the WebDAV approach allows the most appropriate schema to be used in any context.

There are currently two complementary views on how Web metadata can be provided, the attribute-value approach and the link approach. The attribute-value approach views metadata as consisting of (attribute-name, attribute-value) pairs such as ("Organization," "ACM"). The link approach uses a typed (or annotated) hypertext link between the source resource (which the metadata describes) and the metadata itself, which is at the destination of the link. As an example, a link of type "Organization" can be defined on a Web resource, and the destination of this link can be many things: the ACM home page, a resource containing the text "ACM," or an email address for the organization. Complicating the issue is the existence of straightforward transformations between the two approaches: a destination address can be stored in an attribute-value field, thus simulating the link approach using attribute-values, and an attribute-value pair can be simulated using links by storing the attribute name in a link type and the attribute value in the destination resource.

Whether metadata should be stored inside a Web resource (e.g., the HTML LINK tag) or externally is an orthogonal issue. If stored externally, there is a further distinction between packaging metadata items together or storing them individually. If metadata is stored internally, it is easy to copy and move metadata with the resource, and the metadata is preserved even when not accessed via HTTP, e.g., when a resource is copied to a floppy disk. Metadata stored externally to a resource has the advantage that it can be defined on a resource of any media type, especially those that do not allow metadata to be defined internally. External storage of metadata is the best approach for associating metadata with legacy data formats.

In its initial drafts, the WebDAV group proposed that metadata be defined by using the link approach, with bidirectional typed links stored externally to Web resources, but associated with the resources. Consistent with the link approach, the type of link indicates the type of metadata, while the destination of the link contains the value of the metadata. The group is debating the tradeoffs between this approach, which stores metadata individually, and one in which metadata is packaged together. Individual storage of metadata is simpler, and easier to extend to international use domains where a metadata value might be represented in multiple languages. However, packaged metadata makes it easier to set multiple pieces of metadata simultaneously, remove associated metadata when a resource is deleted, and may be more efficient to search.

NAME SPACE MANAGEMENT

In the current publish/browse model of the Web, there is scarce need for a user to duplicate or rename Web resources. However, once the Web is used for distributed authoring, the need for these capabilities, plus the ability to get a listing of a directory, becomes extremely important. Being able to discover what resources currently populate a portion of the name space of a Web server and the ability to copy, move, and delete (already provided in the current HTTP/1.1 specification) these resources form the key elements in managing a Web name space.

There are several justifications for adding copy and move capability. A resource may need to be copied due to changing ownership, prior to major modifications, or when making a backup. It is often necessary to move (i.e., change the name of) a resource; for example, due to adoption of a new naming convention or a typing error in entering the name originally. (Due to network costs associated with loading and saving a resource, it is far more efficient to have a server perform a resource copy or move than for a client to do so, thus preventing the contents from being transmitted twice.)

Defining the meaning of copy and move is difficult in the context of the Web because some classes of Web resources correspond to the output of a computational process, such as a CGI program or script. Since one CGI program can potentially generate an extremely large number of resources, what does it mean to copy one of the outputs of the script? The WebDAV group addressed this issue by adding a "source" link to automatically generated resources, which points to the location of the program that generated the page. Copy and move can then be applied to the source program.

Copy and move also have ramifications with respect to metadata: how should metadata behave after a copy or a move? It seems that all metadata on the duplicated or moved resource should be identical to the metadata on the original. However, there are really two classes of metadata: client controlled and server controlled metadata. Client-controlled metadata is any metadata set by the end-user, such as a comments value. Server-controlled metadata is set by the server, and may vary with location in the name space. For example, on one Web server the cache expiration date (when a Web cache discards and reupdates its copy of a resource) is automatically set to never expire in the directory that only contains bitmap images. If a resource is copied or moved to this directory, the server-controlled cache expiration date is automatically set, possibly overwriting the previous value. Our current approach is that client-controlled metadata should be copied/moved, but server-controlled metadata should not, since it is automatically updated.

Defining the listing of the contents of a directory on a Web server is also difficult because, by definition, the name space created by the http URL scheme is flat. However, since most Web servers use a file system for storage and map the URL name space into the hierarchically ordered name space of the file system, the notion of getting a listing of the contents of these file system directories via the Web is useful. Adding further difficulty is the convention of mapping the URL of a file system directory to the file "index.html," making it difficult to do a straightforward retrieval of directory contents.

The current approach is to create a new media type for hierarchically ordered collections (e.g., file system directories) which groups metadata information such as resource name, last modified date, owner, and size. Exactly how this collection is accessed is not yet determined. One view is to have a hierarchical collection returned in response to read accesses to resources that map to file system directories, while another is to define a new "index" operation that returns a hierarchical collection. A third approach is to have links on every resource pointing to the collection(s) containing the resource. The hierarchical collection can then be retrieved by reading the resource at the destination of the link.

Overwrite Prevention

Once two people start collaborating on the same document, the issue of write control comes to the fore. If everyone can write to the same, unversioned document, then it is possible to lose changes made by one or more contributors, as first one collaborator then another writes changes without first merging in previous updates.

There are many techniques that can be used to alleviate this "lost update" problem, several of the more common are

—POTS (plain old telephone service) or "over-the-wall" control. This scheme is a social convention in which collaborators agree to communicate verbally when one author has finished working and it is safe for another to begin. Email can also be used to implement this write control scheme, as can a physical object (e.g., a baton) passed from author to author in environments where authors are colocated.

—Advisory locks or reservations. In this scheme an author indicates to the computer that controls access to the document that he intends to edit it, and the computer records the author's intent. If another author tries to indicate his intent to edit, the computer will notify them that the document is currently being edited. However, if the second author still wishes to edit, he may do so, presumably contacting the other author to negotiate access, or taking advantage of extra-system knowledge that no conflict will result (e.g., the other author is in a meeting).

—Strict locking. In this scheme an author indicates to the computer controlling access to the document that he intends to modify it, and the computer responds by locking the document. Once the document is locked, it may not be modified by anyone other than the owner of the lock. Other authors who try to edit the same document are refused because they do not own the lock.

These schemes vary from least protective and most flexible (POTS) to most protective and least flexible (strict locking).

The WebDAV approach currently is to provide for strict locking. An exclusive write lock capability prevents two (or more) users from overwriting each other's work. A lock discovery mechanism allows authors to find out if any locks exist on a Web resource. Since the Web is designed so that no lock is required to read a Web page, there is no concept of a read lock. The contents of a resource may change without warning if a write lock is not owned on the resource.

The distributed nature of the Web makes strict locking the best approach for write control on the

Web. Since collaborators can potentially be many time zones apart, POTS locking doesn't work very well. It is hard to acquire the knowledge necessary to determine when it is safe to override an advisory lock. Strict locking guarantees that the document will not be modified by anyone other than the owner of the lock, providing definite assurance that changes will not be clobbered.

Locking usually comes paired with notification, so that other collaborators can be automatically informed by the system when a lock has been released. Due to the client-server nature of the HTTP protocol, the WebDAV group has decided not to provide notification capability, which is a server-to-client communication. Providing such capability could result in the possibility that notification would be sent to a Web client at any time via HTTP, without the client requesting it. This is counter to the current model in the HTTP protocol, where a client makes a request and the server sends information in response, but is consistent with more recent server "push" models. What should happen when a notification needs to be sent but a collaborator isn't online is still unresolved. I expect that notification facilities will be provided by some non-HTTP communications path, perhaps via email.

VERSIONING

Systems to create, manage, and browse significant previous states of a resource are known as version management, versioning, or revision control systems. Versioning systems typically employ a "library" model. When an author wants to begin work on a resource, he checks out the resource, the way a book is checked out at a library. When finished, he checks it in, entering some descriptive comments about the change, and the versioning system freezes and stores the current state of the resource.

A versioning system is collaboration infrastructure. If work on a document needs to proceed with two or more collaborators working simultaneously, a versioning system can be used to split the document into several parallel development paths. Each collaborator has a separate copy of the document to work on, which they can lock, preventing lost updates. At the end, the collaborators use a merge tool to combine their changes together.

A version management system can also be used as infrastructure in the construction of a configuration management system, which manages versioned collections of versioned resources. Configuration management systems are useful for managing change to a Web site, and are also used for managing software projects.

Because they store previous states of a document, versioning systems are useful for browsing the history of a document or viewing its previous states. All versioning systems provide a (sometimes graphical) view of the graph of predecessor and successor relationships between different versions of the same resource, including the author, date, and comments associated with a particular change. Versioning systems also allow the previous states of a resource to be viewed.

Variations among the check-out and check-in behaviors of versioning systems occur in three dimensions:

—Write control. Some versioning systems (e.g., SCCS [Rochkind 1975]) perform strict locking, others have advisory locks (e.g., RCS [Tichy 1985], that can do either strict or advisory locking), and others that do no locking at all (e.g., CVS [Berliner 1990]).

—Setting version identifier. Whether the identifier of a particular version (e.g., "2.1") can be set by the user and whether it is set at check-out or check-in varies.

—Working document location. Where the checked-out, or working, version of the document is located varies, as does whether this working copy of the document is part of the versioning system's repository.

The goal of the WebDAV approach has been to provide versioning operations that support browsing revisions of a piece of information, viewing the revision history of a piece of information, and the ability to perform check-out and check-in. So far, the WebDAV approach has focused on trying to create a simple interface for check-out that accommodates the varying versioning styles described above, so as to support the many different existing versioning systems that might possibly be used as the versioning engine in a Web server. This requires a flexible client adaptable to varying versioning semantics. However, this runs counter to conventional wisdom that holds that clients should be kept simple. As a result, the degree to which different versioning schemes will be accommodated is still being considered.

Conclusion

This article describes a rationale for the existence of a standards body charged with developing an interoperability specification for distributed authoring and versioning on the World Wide Web. It also provides an overview of the capabilities that the WebDAV working group proposes adding to the www.

Working groups of the Internet Engineering Task Force are completely open. If you wish to participate in the discussions on WebDAV topics, you may join the mailing list by sending an email with subject "subscribe" to w3c-dist-auth-request@w3.org. The home page for the WebDAV group is at http://www.ics.uci.edu/~ejw/authoring/, which contains links to current working drafts, email list archives, and background material. **SV**

References

BERLINER, B. 1990. CVS II: Parallelizing software development. In Proceedings of the 1990 Winter USENIX Conference (Washington, DC, 1990).

BERNERS-LEE, T., MASINTER, L., AND MCCAHILL, M. 1994. Uniform resource locators (URL). RFC 1738, CERN, Dec. 1994. ftp://ds.internic.net/rfc/rfc1738.txt

FIELDING, R., GETTYS, J., MOGUL, J., FRYSTYK, H AND BERNERS-LEE, T. 1997. Hypertext transfer protocol—HTTP/1.1 RFC 2068. U.C. Irvine, Irvine, CA, Jan. 1997. http://www.ics.uci.edu/pub/ietf/http/rfc2068.txt

GOLAND, Y.Y., WHITEHEAD, E.J., JR., FAIZI, A., CARTER, S.R. AND JENSEN, D. 1997. Extensions for distributed authoring and versioning on the World Wide Web—WebDAV. Internet-draft, work-in-progress. ftp://ds.internic.net/internet-drafts/draft-jensen-webdav-ext-00.txt

LASSILA, O. 1996. HTTP-based distributed content editing scenarios. Internet draft, work-in-progress. ftp://ds.internic.net/internet-drafts/draft-lassila-http-edit-dist-00.txt

RAGGETT, D. 1997. HTML 3.2 reference specification. W3C Recommendation, REC-html32, Jan. 14, 1997. http://www.w3.org/pub/WWW/TR/REC-html32.html

ROCHKIND, M. 1975. The source code control system. IEEE Trans. Softw. Eng. SE-1, 14 (Dec.), 364–370.

SLEIN, J., VITALI, F., WHITEHEAD, E.J., JR., AND DURAND, D. 1997. Requirements for distributed authoring and versioning on the World Wide Web. Internet-draft, work-in-progress. ftp://ds.internic.net/internet-drafts/draft-slein-www-dist-author-00.txt

TICHY, W.F. 1985. RCS—A system for version control. Softw. Pract. Exper. 15, 7 (July, 1985), 637–654.

WEIBEL, S., KUNZE, J. AND LAGOZE, C. 1997. Dublin Core metadata for simple resource description. Internet-draft, work-in-progress. ftp://ds.internic.net/internet-drafts/draft-kunze-dc-00.txt