WEBDAV: IETF Standard for Collaborative Authoring on the Web

The World Wide Web Distributed Authoring and Versioning working group is extending HTTP1.1 to provide a standards-based infrastructure for asynchronous collaborative authoring on the Web.

E. JAMES WHITEHEAD, JR.
University of California at Irvine
MEREDITH WIGGINS
IEEE Internet Computing

Let's say you are collaborating on an engineering project with one or more groups in geographically distant locations. Chances are you are using e-mail to help manage the project, notifying each other of updates to documents and code, filling directories with (not necessarily well-organized) revisions, sending messages to clarify what is completed and what remains to be done.

What if instead you could simply edit Web documents (or any Web resource) in place at a URL? What if these documents were protected by a discoverable locking mechanism so they could not be overwritten, and you could effectively limit access rights? What if versions could be stored for later retrieval, and easily associated with relevant metadata such as author and last date modified?

The implementation of such a scenario might well change the way we think about the Web as a collaborative medium. Just as static documents have evolved to active documents, for Web authoring may well evolve from the local Web server to the entire Web as a new generation of tools support seamless collaborative authoring.

This is the vision behind the World Wide Web Distributed Authoring and Versioning working group of the Internet Engineering Task Force. WebDAV is extending HTTP to provide a standard infrastructure for asynchronous collaborative authoring across the Internet. The WebDAV extensions support the use of HTTP for interoperable publishing of a variety of content, providing a common interface to many types of repositories and making the Web analogous to a large-grain, network-accessible file system.

Influential vendors such as Microsoft, Netscape, Xerox, IBM, and Novell have assisted in the WebDAV development, as has the World Wide Web Consortium. It is the intent of WebDAV to define the standards to make the Web a truly collaborative environment.

1089-7801/98/$10.00 ©1998 IEEE

WEBDAV EXTENSIONS TO HTTP

Although Tim Berners-Lee developed the first prototype browser with read/write capabilities in 1990, most early Web browsers were read-only. When Marc Andreesen's NCSA Mosaic was ported from X-Windows to the Windows and Macintosh platforms, along with its lib.gif files, the Web exploded on a read-only basis.

The first commercial tools to support remote Web authoring arrived in 1995 with NaviSoft's NaviPress and Vermeer Technologies' FrontPage (which caught the interest of America Online and Microsoft, respectively). From that time on an increasing number of tools, including Web-integrated word-processors and rapid application development environments, began offering capabilities to publish pages to Web servers.

Experience from the initial Web-based authoring systems was clear on one point: No useful Web authoring system could be constructed without extending HTTP. Both NaviPress and FrontPage had to add functionality on top of HTTP to meet user needs. Working in isolation, every tool extended HTTP in a different, noninteroperable way. WebDAV extends HTTP by providing a set of open standards that can be used by all authoring tools.

To date the WebDAV working group has defined a set of extensions to the base Hypertext Transfer Protocol for six capabilities. The first three—overwrite prevention, properties, and name-space management—are fully implemented and described in some detail later in this section. Figure 1 shows how a typical Web client could use these extensions to edit the source of a Web resource.

The other three extensions, scheduled for completion in Fall 1999, are the following:

- *Version management*. Supports the storage of important document revisions for later retrieval. Version management can also support collaboration by allowing two or more authors to work on the same document in parallel tracks. Automatic versioning records successive modifications to a resource made by versioning-unaware ("downlevel") clients.
- *Advanced collections*. Support for advanced resource collections that contain referential members, as well as collections ordered by the client. (In WebDAV, collections provide a mechanism for hierarchically organizing network resources.) Referential resources act like symbolic links in a file system, allowing the resource to be reused in multiple collections. They also allow the collection to contain non-HTTP resources. Ordered collections maintain a client-specified ordering of resources; they are useful, for example, in collections that contain human-ordered contents, such as the chapters of a book.
- *Access control*. Limits the access rights of a given authenticated principal on a given resource. WebDAV assumes that all WebDAV applications will support **HTTP Digest Authentication**, the cryptographic authentication scheme that is part of the HTTP 1.1 protocol.

In addition, in a closely related effort, the DAV Searching and Locating (DASL) group is working to develop an interoperable means of searching a

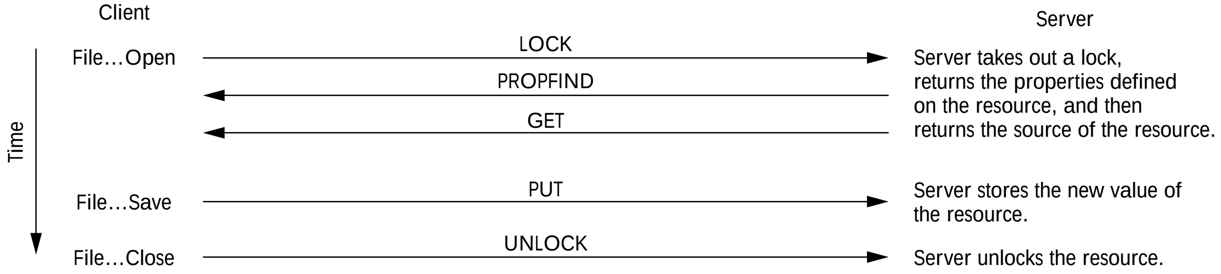
Client
Server
Time
File…Open
LOCK
Server takes out a lock, returns the properties defined on the resource, and then returns the source of the resource.
PROPFIND
GET
File…Save
PUT
Server stores the new value of the resource.
File…Close
UNLOCK
Server unlocks the resource.

Figure 1. Protocol requests that a generic WebDAV client would make to carry out specific client actions. Arrows indicate the predominant information flow associated with a request. All requests use the WebDAV plus HTTP 1.1 protocol. In this scenario, the user of a WebDAV-capable application selects the resource they want to edit using a standard File Open... dialog box. The application then uses WebDAV LOCK to lock the resource and thus prevent modifications by other applications. Next, WebDAV PROPFIND retrieves the resource's properties; and HTTP GET retrieves its contents, which are then displayed for editing. Once all edits have been completed, HTTP PUT saves the resource back to the Web, and WebDAV UNLOCK removes the lock, allowing collaborators to access the resource.

repository that is compliant with the WebDAV object model, and that organizes its resources into URL hierarchies. The main capability of DASL is searching, that is, support for client specified, server-executed queries to locate resources based on their metadata properties and text content.

A more detailed discussion of the capabilities currently supported by WebDAV is included below.

Overwrite Prevention

The issue of write control or locking is crucial in a collaborative environment. When two or more people can write to the same, unversioned document, changes can be lost as first one collaborator, then another makes changes without first merging in previous updates. This is the so-called “lost update problem.”

WebDAV provides an *exclusive write* lock, which guarantees that only the lock owner can overwrite a locked resource, and a *shared write* lock, which allows a group of collaborators to work together on a resource. Locks automatically time out, easing administration of a WebDAV server. By supporting mechanisms for both shared and exclusive locking, WebDAV can accommodate a wide range of collaborations. Shared locks work best in environments in which collaborators are aware of each other’s activities; exclusive locks provide a higher degree of conflict avoidance for collaborators who are not in close contact, or during periods of high contention for a document.

In WebDAV locks may have a scope of a single resource or a hierarchy of resources, such as a collection and its member resources. A lock discovery mechanism (a WebDAV “property”) 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. An implication of this fact in a “writable” Web environment is that the contents of a resource may change without warning if a write lock is not owned on the resource.

Locking usually comes paired with event notification capability, so that collaborators can be notified by the system when a lock is released. Notifications may occur in multiple granularity levels. Events with a grain size of an entire resource, such as a lock being granted or released, provide document access awareness information, while subresource events, such as a word being inserted into a paragraph, can lead to authoring tools that support multiple authors working simultaneously on the same document. Although WebDAV has decided against developing an interoperability standard for Web-based notifications, the **Event Notification Service BOF** meeting held at the Chicago IETF meeting in August 1998 is an indication that standardization work may soon begin in this area.

Properties

All information published on the Web has many additional pieces of information associated with it, such as title, subject, creator, publisher, length, and creation date. This information about information (called properties within WebDAV, but also known as metadata) is particularly useful when searching for Web resources—by focusing a search on the value of a particular property (for example, the author), properties can be used to reduce the number of unrelated query results.

WebDAV properties are *name, value* pairs, where the name is a URL, and the value is a sequence of well-formed Extensible Markup Language (XML). The URL name allows new properties to be added without having to register them centrally, while XML values provide typing via structured markup and extensibility for the addition of new tags. Since all content within XML is encoded between start and end tags, additional elements can be easily added to a property by inserting new tagged content.

Internationalization is another major benefit of XML. Since XML mandates support for the UTF-8 and UTF-16 encodings of the ISO 10646 character encoding standard, as well as language tagging, properties can express content in the vast majority of human languages. Finally, by using XML, WebDAV properties can support other metadata activities based on XML, such as the **Resource Description Framework** (RDF) under development at the W3C.

The development of useful sets of properties, or metadata, is considered by most Web authorities to be extremely important for the future of the Web considered as both a human- and machine-readable resource. One set of metadata developed to assist Web searching is known as the **Dublin Core**, a schema for the creation of digital library catalogs. Since the WebDAV working group expects natural communities like the users of the Dublin Core to develop their own metadata sets, WebDAV decided to focus instead on developing facilities for creating, modifying, deleting, and retrieving metadata.

These facilities allow for the manipulation of metadata from multiple schemas, allowing the schema itself to vary with domain of use. While the

Dublin Core schema, for example, is appropriate in a general context, it may not be ideal in a legal setting. By being metadata-schema–neutral, the WebDAV approach allows the most appropriate schema to be used in any context. WebDAV thus focuses on *how* properties are stored and retrieved, rather than on *what* properties mean.

The DAV Searching and Locating (DASL) effort, a follow-on effort to WebDAV, is working on a standard mechanism for searching WebDAV servers. DASL dramatically increases the value of properties, providing an efficient way to search for properties that match client-specified criteria. For example, when DASL is paired with a bibliographic standard like the Dublin Core, searches can be made across a server for resources by a specific author, or by that author between certain dates. Unlike search engines like AltaVista, DASL searches are performed directly by the server that contains the resources, and hence will not return stale values.

Name-Space Management

If the Web is to successfully support distributed authoring, users must be able to manage the name space of the Web server they are writing to. Discovery of what resources currently populate the relevant portion of the server name space, plus the ability to copy, move, and delete those resources, are the key elements required.

In WebDAV, collections provide direct containment for local resources and referential containment for resources located anywhere on the Web. The WebDAV server maintains consistency in the direct containment relationships. For example, when a new resource is PUT into the namespace of a collection, it is automatically added to the collection.

WebDAV offers the ability to copy and move Web pages and to receive a listing of resources in a collection. *Copy* capabilities offer support for changes in resource ownership, as well as resource modifications and backups. *Rename* capabilities support the adoption of new naming conventions or the correction of typing errors.

It turns out, however, that defining the meaning of copy and rename can be problematic in the context of the Web, because some classes of Web resources, such as CGI programs or other scripts, correspond to the output of a computational process. Since one CGI program can generate an extremely large number of resources, what does it mean to copy one of the outputs of the script? WebDAV has addressed this issue by adding a "source" link to automatically generated resources that points to the location of the program that generated the page. Copy and rename can then be applied to the source program.

Copy and move also have ramifications with respect to properties. How should properties behave after a copy or a move? It would seem that all properties on the copied or moved resource should be identical to those on the original. However, there are really two classes of properties: live and static. Static properties retain their value, once set, until they are explicitly modified by a client. Live properties, in contrast, have their syntax and semantics enforced by the server, and may vary at any time. One example of a live property is the content length of a resource: Each time the resource is updated, the value of the property will also be updated.

WebDAV provides both an exclusive-write and a shared-write lock for overwrite protection.

Live properties may apply to only part of a server's namespace. For example, on one Web server the cache expiration date (when a Web cache should discard and re-update its copy of a resource) is automatically set to never expire only in the directory containing 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. These cases are addressed by having copy and move operations perform a best-effort duplication of properties, duplicating live properties at the destination when possible, and replacing them with a static property containing a snapshot of the live property's value if not. Optionally, if best-effort is too uncontrolled, copy and move can be instructed to fail when live properties cannot be maintained as such at the destination.

Listing the contents of a collection is very similar to listing a directory of a file system. Most existing operations for listing a directory, such as "ls" or "dir," provide the name of a file, and an option for retrieving limited sets of properties about the file, such as its size, owner, and access permissions. Because WebDAV has an existing property-retrieval mechanism, it made little sense to define a new operation for listing a collection. WebDAV's property retrieval allows, with a single operation, a hierarchical retrieval of properties on a collection, returning for each resource its name and requested properties. This mechanism had enough expressive power to serve as the “list a collection’s members” operation as well.

HOW WEBDAV EXTENDED HTTP

There are three general mechanisms available for adding functionality to HTTP: URL “munging,” RPC-via-POST, and adding new methods.

URL Munging

URL munging is a technique whereby a command is appended to the end of a URL, typically after a “?”. For example, a versioning system might add a checkout command to a URL as follows: http://www.editing.com/myresources/index.html?command=checkout.

URL munging has the advantage of being easily parsed by a CGI script. The script looks for the question mark, and then parses everything after it, and acts as the intermediary between HTTP and the underlying repository (in this example, the store used by the version control system). However, there are also significant drawbacks. The first problem is interactions between different URL munging schemes. If two extensions use the same URL munging convention, it will cause adverse interactions between them.

WebDAV shows promise as an infrastructure for distributed software engineering.

An even worse problem is the assumption by URL munging that HTTP GET is the only method used to retrieve Web contents. Unfortunately, the semantics of HTTP’s other methods become unclear when applied to a munged URL. Because of these problems, URL munging is widely disparaged within the community of protocol developers extending HTTP.

HTTP-via-POST

Remote procedure calls permit a client to execute a procedure on a remote server. This is accomplished by marshalling parameters for the operation on the client side, and sending the parameters, along with an identifier for the function to be executed, to the server. The server unmarshals the parameters, executes the specified function, and then sends back results to the client (after a marshalling step).

In the RPC-via-POST method of extending HTTP, a client marshals parameters into an HTTP message body, along with an identifier for the function to be performed, and then sends the message body to the server using the HTTP POST method. POST is used because it has a sufficiently open definition so that almost any operation can be invoked using it. The server performs the stated operation and returns a message body in the response giving the results of the operation. The Internet Printing Protocol is an example of a protocol that uses this approach.

The advantage of RPC-via-POST is that, like URL munging, it easily maps onto existing mechanisms for extending HTTP servers such as CGI, Microsoft’s Internet Server API (ISAPI), and the Netscape Server API (NSAPI). It also allows domain-specific marshalling of parameters, so that all parameters do not need to be mapped into HTTP methods and headers. The latter has the benefit of reducing unforeseen interactions with the rest of HTTP’s operations. The drawback of this approach is that POST ends up being a security hole through which almost any operation can be executed. Since each individual extension is free to marshall its parameters in separate ways, a program trying to determine what operation is being performed by looking at a POST message body would have a very difficult task indeed. For similar reasons, performing access control on POST-based operations is extremely difficult.

Adding New Methods

The final mechanism for extending HTTP is to create a new HTTP method for each new operation (which in practice translates into using the HTTP headers to marshall parameters for each new operation). The “add new methods” approach has the advantage of using the designed-in mechanism for extending HTTP, which can therefore take advantage of existing features such as operation precondition headers like If-[None-]Match. Operation-based security and access control is easy, since the operation always occurs at a predictable location in the protocol stream (the HTTP method is always at the beginning of an HTTP message).

There are some drawbacks, however. First, it turns out that HTTP headers are not sufficient for marshalling many kinds of parameters. For example, data that is potentially unbounded in length cannot be well-encoded in HTTP headers, nor can data that contains human-readable text (which requires internationalization using multi-octet character sets like Unicode, along with language tagging). Second, since existing HTTP headers can be applied to any new method, the interactions between existing headers and new methods need to be explicitly defined. Finally, the existing mechanisms for extending HTTP servers do not easily accommodate adding new methods.

However, in the end, the security and access control advantages of adding new methods outweighed the (relatively minor) drawbacks of the approach. In those cases in which parameters could not be easily marshaled into HTTP headers, WebDAV has used XML in the message body to encode the parameters, thus gaining some of the advantages of the RPC-via-POST approach without its security and access control disadvantages.

PROTOTYPE ACTIVITY

Several organizations are developing WebDAV prototypes (see the sidebar, "Further Information on WebDAV," for URLs). Microsoft has included support for WebDAV in the Internet Information Server, which ships with Windows NT Server 5.0 Beta 2. IBM, Netscape and Xerox PARC have all created prototype WebDAV servers. Two research-based efforts include a Java-based prototype WebDAV server from Columbia University, and a Java-based WebDAV Explorer client, providing a Windows Explorer-like interface for a WebDAV repository, from the University of California at Irvine. These prototypes provided valuable feedback during the development of the WebDAV protocol, ensuring the protocol document is concretely specified, and implementable.

Based on this prototype experience, the protocol is a solid specification, ready now to be used in a wide range of applications.

CONCLUSION

To date, the Web has been a read-only medium. WebDAV changes this, making the Web a writable, collaborative medium. What should you expect from this transformation?

Initially, WebDAV will be supported by existing tools that operate on one file at a time. Such tools—word processors, spreadsheets, and graphics and text editors—will still support operations on files in a file system but will additionally support "save to the Web" capabilities. This will open the door for remote collaboration, supporting a seamless transition from individual to collaborative group work.

WebDAV will have its greatest initial impact on small- to medium-sized workgroups that homogeneously support distributed authoring and versioning. Because documents will be editable in-place on the Web, organizations will begin to see greater benefit in performing work in the Web environment rather than on local, isolated disk drives. This will in turn provide an incentive for work practices to coalesce around a local intranet. Over time, as critical mass grows, WebDAV will also reduce the accidental costs of collaboration between workgroups and between organizations. Once documents are routinely kept on a local intranet, starting a collaboration with another group in the same organization, or across organizational boundaries, becomes a simple matter of adding another person to the access control list for the document.

WebDAV also shows promise as an infrastructure for developing distributed software engineering environments and other complex information products. Since the artifacts of software development, like requirements documents and source code, are amenable to remote authoring via WebDAV, distributed authoring and versioning can support virtual teams working together on a common software development task.

Finally, WebDAV will make it much easier for the average person to create Web-pages, since they will be editable in-place, typically in a location provided by an Internet service provider. Contrast this

URLs for this report

An Extension to HTTP : Digest Access Authentication, RFC 2069 • ftp://ftp.isi.edu/in-notes/rfc2069.txt

Dublin Core Metadata • purl.org/metadata/dublin_core/

Internet-Scale Event Notifications • www.ics.uci.edu/pub/ietf/notify/

W3C's Resource Description Framework • www.w3.org/RDF/Overview.html with the current situation, where users have to FTP files to a specific directory, and understand how these files map into a set of URLs.

Product support for WebDAV is just beginning to appear. The next few years should see strong growth in the development of Web-based collaborative tools. Enhanced collaboration will help the Web achieve its promise as the most flexible medium of world-wide communication yet seen. ■

E. James Whitehead, Jr., is a PhD student in the Department of Information and Computer Science at the University of California, Irvine. His research focuses on hypertext and software architecture. Whitehead has been the Chair of the Internet Engineering Task Force World Wide Web Distributed Authoring and Versioning (WebDAV) working group from its inception to the present. Professional experience includes a position at the Raytheon, where he wrote firmware in C and Ada for the German Civilian Air Traffic Control System (DERD) and for a prototype Microwave Airplane Landing System.

Meredith Wiggins is on the staff of *IEEE Internet Computing*, where she serves as acquisitions editor.

Readers may contact Whitehead at Information and Computer Science, University of California, Irvine, California 92697-3425; e-mail ejw@ics.uci.edu.

FURTHER INFORMATION ON WEBDAV

The World Wide Web Distributed Authoring and Versioning working group began in December 1995 as an informal meeting of people interested in remote authoring at the World Wide Web conference in Boston. WebDAV was formally confirmed as a workgroup within the IETF in the Spring of 1997.

Work thus far has focused on three documents:

- a *scenarios document*, which gives a series of short descriptions of how distributed authoring and versioning functionality can be used (HTTP-based Distributed Content Editing Scenarios, http://www.ics.uci.edu/pub/ietf/webdav/scenarios/draft-ietf-webdav-scenarios-00.txt),
- a *requirements document*, which describes the high-level functional requirements for distributed authoring and versioning (Requirements for a Distributed Authoring and Versioning Protocol for the World Wide Web, ftp://ftp.isi.edu/in-notes/rfc2291.txt), and
- a *protocol specification*, which describes new HTTP methods, headers, request bodies, and response bodies to implement the distributed authoring and versioning requirements (Extensions for Distributed Authoring on the World Wide Web—WebDAV, http://www.ietf.org/internet-drafts/draft-ietf-webdav-protocol-08.txt).

The DAV Searching and Locating group (DASL), a related effort, is currently in-process of becoming an IETF working group, and is developing extensions to the WebDAV Distributed Authoring Protocol specification (and hence to HTTP) for searching WebDAV repositories. DASL has its own requirements document (http://egg.microsoft.com/dasl/files/draft-dasl-requirements-00.txt) and protocol document (ftp://ftp.isi.edu/internet-drafts/draft-reddy-dasl-protocol-02.txt), which are still the subject of intense effort within the DASL group.

The WebDAV and DASL Mail Lists

Information about the IETF WebDAV working group, including mailing list archives and instructions on joining the group is available at **http://www.ics.uci.edu/pub/ietf/webdav/.**

Similar information for the DASL group is found at **http://www.ics.uci.edu/pub/ietf/dasl/.**

URLs for WebDAV Prototypes

Columbia University prototype WebDAV server • http://www.cs.columbia.edu/~eaddy/webdav.html
Microsoft's NT Server 5.0 • http://www.microsoft.com/windowsnt5/server/overview/InfoSharing.asp
UC Irvine WebDAV Explorer client • http://www.ics.uci.edu/~webdav/

Key WebDAV Participants

The WebDAV specifications have been developed by a core group of participants. Authors on WebDAV specifications are:

- Alan Babich, FileNet;
- Steven Carter and Del Jensen, Novell;
- Jim Davis and Judith Slein, Xerox;
- David Durand, Boston University;
- Asad Faizi, Xenosys;
- Yaron Goland, Rick Henderson, Alex Hopmann, Chris Kaler, Lisa Lippert, and Saveen Reddy, Microsoft;
- Surendra Reddy, Oracle;
- John Strake, Netscape;
- Fabio Vitali, Univ. of Bologna;
- Jim Whitehead, UC Irvine.

Other contributors include Larry Masinter, Xerox; Jim Amsden, IBM; John Turner, PC Docs; Mark Day, Lotus; and Richard Taylor and Roy Fielding, UC Irvine.