How to implement Web-based Groupware Systems based on WebDAV [*]

Fredj Dridi, Gustaf Neumann
Information Systems and Software Techniques
University of Essen, Universitätsstraße 9, D-45141 Essen, Germany
{*Fredj.Dridi,Gustaf.Neumann*}*@uni-essen.de*

Abstract

The protocol WebDAV (World Wide Web Distributed Authoring and Versioning) is an extension of HTTP/1.1 and defines new methods for supporting collaborative work (primarily asynchronous collaborative authoring) on the Web. The new WebDAV methods provide a good basis for building Web-based collaborative applications. In order to use the new WebDAV functionality, a client application must have direct access to the protocol methods and must be able to pass certain parameters provided by the user. Common Web browsers provide only limited support for accessing HTTP methods and no support for accessing header fields directly.

This paper presents a generic approach for implementing WebDAV-based applications by extending HTML forms to provide direct access to the protocol methods (HTTP and WebDAV methods) and the header fields from within a Web browser. This approach is used to develop a simple collaborative project repository based on WebDAV.

1. Introduction

In general, the World Wide Web has a high potential as a platform for distributed groupware systems. Authors create and change documents locally and upload them to a Web server, where they can be accessed by their collaborators. However, Web browsers and servers which implement the protocol HTTP are mostly limited to provide reading access to Web documents. For this reason, existing Web-based groupware systems (e.g. the BSCW system) are implemented using client/server side scripting. These scripts are used to implement the basic operations needed for this kind of applications.

The protocol WebDAV (World Wide Web Distributed Authoring and Versioning) which was developed as an extension of HTTP/1.1 by the WebDAV working group of the Internet Engineering Task Force (IETF) and defines new methods which offer additional services for editing and managing files on remote Web servers in a structured way (e.g. locking, version management, metadata management, etc.). When using WebDAV, the Web is becoming a rich infrastructure for collaborative applications, where the WebDAV methods can be used to implement the basic operations needed (e.g. copy, delete, upload, assign metadata to documents, etc.). Scripting can be used in this context as well, but not for implementing the basic tasks of collaboration, but to simplify maintenance of the application.

The paper is structured as follows: In Section 2 we introduce the functionality of WebDAV. In Section 3, we briefly summarize related work in the area of implementing Web-based groupware systems and we give a short description of two systems, BSCW and DReSS. Furthermore, we present the advantages of using WebDAV for these kind of applications. Section 4 identifies the requirements of a Web browser user interface to use the protocol WebDAV and defines the implemented extensions within the Web browser "Cineast". Section 5 describes a simple collaborative application using both the WebDAV-compliant Web browser and Web server (e.g. the Apache Web server extended with the module `mod_dav` which implements the WebDAV server side). This application is roughly comparable with the core functionality of the BSCW system and supports a project repository within an organization where the users want to share project related documents and folders. Some features of BSCW (e.g. user administration, group management, awareness information) are not yet implemented in our project repository. Section 6 offers concluding remarks and a brief description of future work.

[*] Published in: Proc. of WETICE 99, IEEE 8th Intl. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Stanford, CA, June 1999

2. Extended HTTP: WebDAV

WebDAV (World Wide Web Distributed Authoring and Versioning) [9] is a conservative extension of the protocol HTTP/1.1 [5] and offers additional services which allow users to update files safely on remote Web servers. WebDAV provides a standard infrastructure for asynchronous collaborative authoring across the Internet in order to turn the Web into a collaborative environment. The core features of WebDAV are [23, 9]:

- *Metadata management.* Metadata (Properties) denote the information associated with Web resources, such as the title, creator, etc. A property is a name/value pair, where the name is a URL, and the value is a sequence of well-formed XML elements [3]. The protocol offers the ability to create, remove, and query properties by using the new methods `PROPPATCH` and `PROPFIND`.
- *Name space management.* WebDAV defines two new methods `COPY` and `MOVE` to instruct the server to copy, move, or delete Web resources. Deleting a resource is already provided in the HTTP/1.1 specification.
- *Collections.* The base WebDAV protocol provides basic support for collections. A collection is like a directory in a file system. WebDAV defines the `MKCOL` method for creating collections and specifies how other HTTP and WebDAV methods interact with collections.
- *Overwrite prevention.* The WebDAV extensions to HTTP provide locking mechanisms in order to prevent resource modifications by other applications (“lost update” problem). The protocol defines therefore two new methods: `LOCK` and `UNLOCK`.
- *Version management.* This component supports the storage and the management of document revisions for later retrieval [10].
- *Access Control.* Since resources may be accessible by multiple applications, an access control system is needed to control how these applications can access and alter a resource. This system defines the access rights of a given authenticated application on a given resource [14]. For authentication purposes all WebDAV applications should support the HTTP Digest Authentication scheme [7].

WebDAV extends HTTP/1.1 by new methods that require certain parameters. In HTTP/1.1 the parameters for methods are exclusively encoded in the HTTP header fields. Unlike HTTP/1.1, some WebDAV method parameters are encoded as XML [3] request bodies. These parameters are mostly application specific and must be provided by the user. Therefore, the use of WebDAV requires either new Web clients or extensions of existing Web browsers which support the additional functionality. In this paper we present an extension of an existing Web browser to demonstrate how to use WebDAV within a Web browser.

3. Web-based Groupware Systems

A Web-based groupware system is an application that allows several users located on different Web sites to cooperate and produce documents in a structured way. This section gives a short description of two Web-based groupware systems and motivates the use of the extended HTTP protocol (WebDAV) in this context.

The two most important purely Web-based groupware systems are the BSCW [2] and the DReSS [1] systems:

- The *BSCW* System (Basic Support for Cooperative Work) provides basic functionality for group cooperation and uses the Web as its communication infrastructure. The BSCW system is based on the metaphor of a “shared workspace”. The system provides an easy-to-use Web interface based on HTML and supports, for example, uploading of various types of objects to a shared workspace. The information sharing functionality is enhanced with basic facilities for visualization of changes, authentication, authorization, and version control. The BSCW system is implemented using a set of CGI scripts and a set of client side scripts.
- *DReSS* (Document Repository Service Station), is a system to enable authors to deposit (and update) documents on a Web server. DReSS (version 2.0) is implemented via client side scripts and server CGI scripts written in Java. DReSS supports single and multi file upload by using the form-based file upload protocol as defined in RFC 1867 [15].

Table 1 summarizes the basic functionality of the *BSCW*, *DReSS*, and the *WebDAV-based Project Repository* introduced in this paper.

As mentioned in [23], the intention behind WebDAV is to change the Web into a collaborative medium, and to provide a common interface to many types of repositories in order to change the Web into the analogous to a large-grain, networked file systems. The new WebDAV methods used to access and provide information on the Web can be used in a meaningful manner to build Web-based collaborative applications. These methods offer the basic support for all operations needed by a Web-based groupware system. Therefore, there is no need for large/complex server and client side scripting. Scripting is just needed to simplify the usage rather than to implement basic groupware operations.

The client side implementation of WebDAV described in this paper provides access to the core WebDAV features in a generic way and is based on an extension of an existing Web browser (see Section 4). There are WebDAV clients which are either specialized Web client applications (e.g. Sitecopy [18]), or do provide limited access to WebDAV

	BSCW	*DReSS 2.0*	*WebDAV-based Project Repository*
File upload	RFC 1867 and/or special client side “helper” application	RFC 1867	PUT
Authentication and Access Control	Server dependent authentication + BSCW specific group-based authorization [20]	Server dependent	HTTP Digest Authentication [7] or WebDAV ACL protocol [13]
Version Control	BSCW specific	–	WebDAV Versioning [10]
Metadata Management	BSCW specific	–	PROPFIND and PROPPATCH
Implementation	CGI + optional Java scripting	CGI scripting + client side helper app.	via protocol methods

Table 1. Comparison of Web-based Groupware Systems

features (e.g. Internet Explorer 5) compared to the functionality presented here.

4. Integrating WebDAV into a Web Browser

In this section, we discuss how to extend a Web browser in a generic way in order to access protocol methods (e.g WebDAV methods). We identify, which kind of interaction with the user at the protocol level is needed to initiate a protocol request. Finally, we propose extensions to HTML forms to improve this kind of interaction between the user and a Web browser.

These extensions are implemented in the extensible Web browser “Cineast” [12] which is a freely available, extensible Web browser. It provides an environment for prototyping new client side (and server side) Internet technologies. The browser is implemented in the Wafe [16] environment which includes support for several libraries like OSF/Motif, XOTcl [17], SSLeay [25], LDAP [24], etc. The Cineast user interface is implemented using OSF/Motif and the Kino widget [11] for HTML (and XML) parsing and rendering. The application logic and the general networking functionality (HTTP/1.1, FTP, etc.) are implemented using the object-oriented scripting language XOTcl, which is an extension of OTcl [22] containing language support for dynamic aggregations, per-object mixins, and design patterns. Cineast has built-in support for HTML forms.

4.1. Requirements for a Web Browser User Interface for WebDAV

Web browsers are primary designed to retrieve Web resources rather than provide direct access to the protocol methods. As mentioned above a WebDAV request requires a set of parameters which should be submitted in form of request header fields and/or request bodies. The input of all needed parameters can not be completely automated and requires user input. For example, to copy a Web resource, the user should provide at least the destination URI and an overwrite flag which indicates whether the source resource should overwrite the destination resource. Generally, three types of user input are needed to initiate a protocol request:

- *Request URI.* A request URI identifies the location of the resource addressed by a protocol method.
- *Request header.* In addition to the header fields defined in the protocol HTTP [5], WebDAV defines new header fields which are necessary to store some parameters needed by a method (e.g. destination for the `COPY` method).
- *Request body.* Some WebDAV methods need a request body which is always a well formed XML element [9]. A request body is used to store some parameters needed by a method (e.g. name of properties to be retrieved by the `PROPFIND` method).

The Web browser user interface provides an input field for typing the request URI, but provides no means of specifying the other two parameter types.

4.2. Accessing WebDAV Methods within a Web Browser via Forms

Generally, a protocol method addresses a target object (Web resource) and requires parameters which are protocol specific or application specific (see Figure 1). From the Web browser point of view, while creating the protocol request the target object is the request URI, the values of

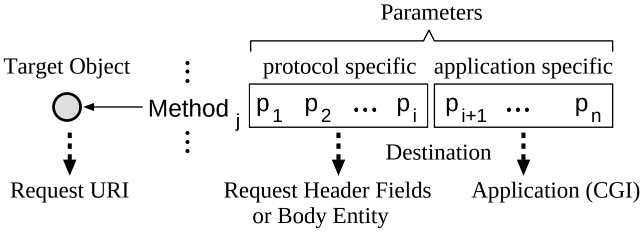


Figure 1. Protocol Request: Target Object, Method, Parameters

the protocol specific parameters are passed in form of request header fields or in the request entity body. The values of the application specific parameters are not necessary for starting the protocol request and are passed to the server side application in a predefined way (e.g. URL-encoded). The focus of this paper is on the protocol specific parameters and how to obtain them from the user. Traditionally, user inputs in form of freely specified values are handled by HTML fill-out forms that have a strictly predefined semantic as defined in [19]. The HTML form definition is limited to only two HTTP methods (GET and POST) and does not provide any means of using selected form input fields as method parameters.

Extending the FORM element

The HTML specification [19] allows to specify the HTTP method via the attribute `method` of a `FORM` element, but is restricted to the two values GET and POST. In addition, the attribute `action` may specify a server side "form processing agent" that implements the action of the form, typically via CGI. However, this capability is limited because the user agent (Web browser) forwards all form data to a processing agent and HTML forms do not provide a way to tell the user agent the destination (protocol or application) of the user input. For example, a user agent should be able to take into account the protocol specific input data while creating a protocol request (e.g. WebDAV request). For this purpose this paper proposes three extensions to HTML forms to overcome these limitations:

- Extend the definition of the attribute `action` of the `FORM` tag to specify the target object (Web resource) addressed by the protocol method.

  ```
  action = uri
  ```

- Extend the values of the attribute `method` of the `FORM` tag with all methods of WebDAV.

  ```
  method = get | post     | put       | delete |
          mkcol | propfind | proppatch | copy   |
           move | lock     | unlock
  ```

- Add a new attribute (`usage`) to the `INPUT` and the `TEXTAREA` tag. This attribute can be used by the user agent to specify the destination (protocol or application) of the user input (see Figure 1).

  ```
  usage = target|method|header|body|application
  ```

  The content of the `INPUT` or `TEXTAREA` tag is used differently depending on the value of the `usage` attribute:

  - `target`: the content is used as the target object and replaces the target object specified by the `action` attribute of the `FORM` element.
  - `method`: the content denotes the protocol method and replaces the method specified by the `method` attribute of the `FORM` element.
  - `header`: the content is included in the header fields of the protocol request.
  - `body`: the content is included in the entity body of the protocol request.
  - `application`: the content is passed to the server side application.

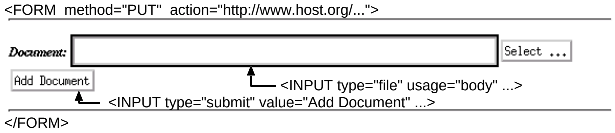

<FORM method="PUT" action="http://www.host.org/...">
<INPUT type="file" usage="body" ...>
<INPUT type="submit" value="Add Document" ...>
</FORM>

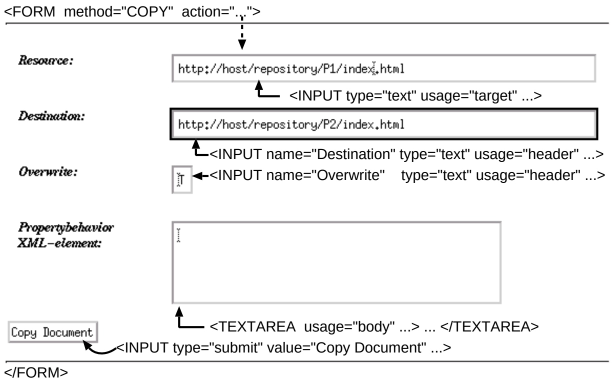
Figure 2. An HTML `FORM` for a PUT Request

Figure 2 depicts an HTML form to specify an HTTP/WebDAV PUT request in order to upload a single file by using the input type `file` defined in RFC 1867 [15]. When the submit button `Add Document` is pressed, the user agent creates an HTTP PUT request as indicated in the attribute `method` and passes the content of the selected file in the request body (`usage="body"`). The attribute `action` indicates the location (target) where to put the new resource. Note, that although the PUT method is a standard HTTP/1.0 method, the commonly used Web browsers provide no standard means to address this method.

Figure 3 shows an HTML form to ask the user for the needed parameters for a WebDAV COPY request. This form consists of three text `INPUT` tags, a `TEXTAREA` tag, and a submit button. The value of the first input field is considered to be the location of the source resource (`usage="resource"`). Furthermore, this value is treated by the user agent as a new value of the attribute `action`. The values of the second and third input fields are designated to be the values of the header fields `destination` and `overwrite` of the COPY request (`usage="header"`). Other request header fields could be supplied in the same way. The value of the multi-line text input should be an XML element [9] and is designated to be included in the COPY request body (`usage="body"`).

Finally, when the submit button "`Copy Document`" is pressed, the user agent creates a WebDAV COPY request with the given header fields and the body data:

```
COPY /repository/P1/index.html HTTP/1.1
Host: localhost
Destination: http://host/repository/P2/index.html
Overwrite: T
```



<FORM method="COPY" action="...">

Resource: http://host/repository/P1/index.html

<INPUT type="text" usage="target" ...>

Destination: http://host/repository/P2/index.html

<INPUT name="Destination" type="text" usage="header" ...>

Overwrite: T

<INPUT name="Overwrite" type="text" usage="header" ...>

Propertybehavior XML-element:

<TEXTAREA usage="body" ...> ... </TEXTAREA>

Copy Document

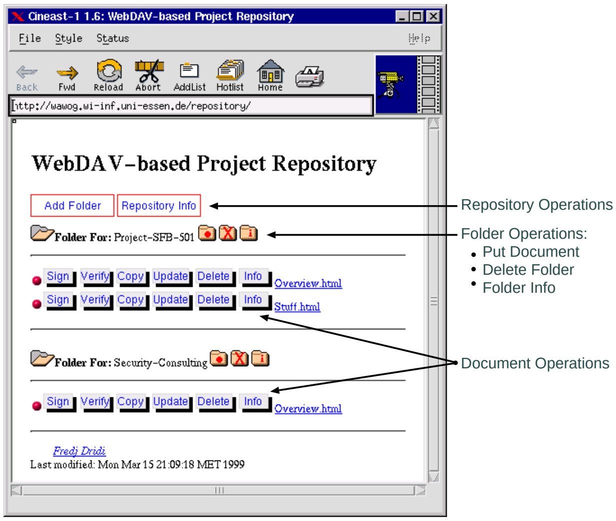
<INPUT type="submit" value="Copy Document" ...>

</FORM>

Figure 3. An HTML Form for a COPY Request

5. Application: WebDAV-based Project Repository

In this section, we describe a simple (lightweight) Web-based collaborative application, where (hypermedia) documents can be shared collaboratively in a controlled way in a project repository. This application demonstrates how to use WebDAV to implement a Web-based groupware system. For demonstration purposes, we have kept this application deliberately small as well as extensible.



Cineast-1 1.6: WebDAV-based Project Repository

http://wawog.wi-inf.uni-essen.de/repository/

WebDAV-based Project Repository

Add Folder | Repository Info

Folder For: Project-SFB-501

● Sign Verify Copy Update Delete Info Overview.html

● Sign Verify Copy Update Delete Info Stuff.html

Folder For: Security-Consulting

● Sign Verify Copy Update Delete Info Overview.html

Fredj Dridi

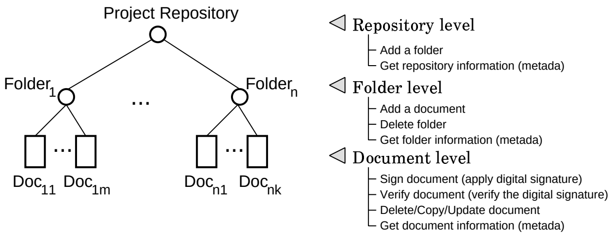
Last modified: Mon Mar 15 21:09:18 MET 1999

Figure 4. Client View of the Project Repository

As a WebDAV-compliant Web browser we used the Web browser “Cineast” [12] (see Figure 4). On the server side we used the Apache Web server [6] in combination with the `mod_dav` module [21] which implements WebDAV. Optionally, we use the cryptographic support from the toolkit `SSLeay` [25], to establish a secure communication via SSL [8] between the server and clients.

5.1. Functionality and Structure

The project repository offers uploading and sharing of documents among varying groups of authors. It is installed in the persistent storage of a WebDAV-compliant Web server (e.g. Apache) and consists of sets of folders (directories). Each folder is a container of topic-related (hypermedia) documents (files). We distinguish between three levels of operation which can be performed on the project repository and the associated metadata set (see Figure 5).



Project Repository
Folder$_1$
Folder$_n$
...
Doc$_{11}$
Doc$_{1m}$
Doc$_{n1}$
Doc$_{nk}$
Repository level
Add a folder
Get repository information (metada)
Folder level
Add a document
Delete folder
Get folder information (metada)
Document level
Sign document (apply digital signature)
Verify document (verify the digital signature)
Delete/Copy/Update document
Get document information (metada)

Figure 5. Functionality and Structure of the WebDAV-based Project Repository

One important new functionality of WebDAV is to assign metadata to Web documents. Metadata are additional information (e.g. creator name, last modified date, security information, etc.) about Web documents. The users of the project repository are able to define and query metadata about the repository itself, the folders, and the documents using the WebDAV methods which are accessible from the browser via HTML forms (see Section 4). For a flexible and scalable metadata management, it is recommended to store metadata in an LDAP-enabled directory [24].

5.2. Implementation

The project repository is viewed and accessed by the Web browser (see Figure 4). The functionality (see Figure 5) is accessed from the Web browser (Cineast) for example by push buttons and is implemented using WebDAV methods. As explained in Section 4.2, WebDAV methods can be accessed within a Web browser via HTML forms. When a user presses a push button on the repository page (e.g. Copy file) an HTML form for the corresponding WebDAV method with appropriate default values is displayed. In order to automate the creation of these HTML forms server (or client) side scripting can be used.

Signing a document or verifying the digital signature of a document can not directly be mapped to WebDAV methods. This can be achieved by combining some WebDAV methods. For example, to sign a document we could first retrieve (GET), sign (using a local cryptographic library), and upload (PUT) the document back to the Web server.

The Cineast Web browser implements HTTP/1.1 and has built-in support for HTML forms. The effort taken to extend the original Cineast browser with the features described in this paper (new WebDAV methods and HTML form extensions) is minimal. We have added about 40 lines XOTcl [17] code in oder to implement the new WebDAV methods and about 70 lines for the HTML form extensions defined in Section 4.2.

6. Conclusion

In this paper we have focused on how to define and implement a Web-based groupware system using the standard protocol WebDAV (World Wide Web Distributed Authoring and Versioning). This protocol offers a basic set of methods which can be used to implement a Web-based groupware system in a structured way. One of the advantages when using WebDAV is that there is no need for large/complex server and client side scripting. Scripting is just needed to simplify the usage rather than to implement groupware basic operations. In order to access WebDAV methods within a Web browser, we have defined and implemented a generic approach where HTML forms are extended. Finally, we have presented a simple collaborative application as a proof of concept. The WebDAV-based project repository establishes a good basis for exploring many interesting questions in the areas of access control and versioning in a collaborative environment. The implementation shown offers a unique test-bed for these emerging technologies.

The future work will focus on the integration of version management and access control aspects. The WebDAV group is working on defining new protocol methods concerning these aspects [10, 13, 14]. Furthermore, we are planning to design a “security management” component in order to administrate a large number of complex hyperlinked documents/metadata and a large number of collaborators with differentiated security requirements [4].

References

[1] A. Aerts, P. D. Bra, and M. Timmermans. DReSS 2.0: Lightweight groupware for hypertext publishing on the web. In *Proceedings of WebNet 98 – World Conference of the WWW, Internet & Intranet*, volume 1, Orlando, Florida, November 1998.

[2] R. Bentley, W. Appelt, U. Busbach, E. Hinrichs, D. Kerr, K. Sikkel, J. Trevor, and G. Woetzel. Basic support for cooperative work on the world wide web. *International Journal of Human Computer Studies: Special issue on Novel Applications of the WWW*, Spring 1997.

[3] T. Bray, J. Paoli, and C. M. Sperberg-McQueen. Extensible markup language (XML) 1.0. *World Wide Web Consortium Recommendation*, February 1998. http://w3c.org/TR/1998/REC-xml-19980210.

[4] F. Dridi and G. Neumann. Towards access control for logical document structure. In *Proc. of the Ninth International Workshop of Database and Expert Systems Applications*, pages 322–327, Vienna, Austria, August 1998.

[5] R. Fielding, J. Gettys, J. Mogul, H. Nielsen, and T. Berners-Lee. Hypertext transfer protocol – HTTP/1.1. *RFC 2068, Standards Track*, January 1997.

[6] R. Fielding and G. Kaiser. The Apache HTTP Server Project. *IEEE Internet Computing*, 1(4):88–90, July/August 1997. http://www.apache.org/.

[7] J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink, and L. Stewart. An extension to HTTP: Digest Access Authentication. *RFC 2069, Standards Track*, January 1997.

[8] A. Freier, P. Karlton, and P. Kocher. The SSL protocol version 3.0. *Netscape*, 1996. http://netscape.com/eng/ssl3/.

[9] Y. Goland, E. Whitehead, A. Faizi, S. Carter, and D. Jensen. HTTP extensions for distributed authoring – WEBDAV. *RFC 2518, Standards Track*, February 1999.

[10] C. Kaler, J. Amsden, G. Clemm, B. Cragun, D. Durand, B. Sergeant, and E. Whitehead. Versioning extensions to WebDAV. *IETF Internet Draft*, January 1999.

[11] E. Köppen and G. Neumann. Implement, Gustaf! the Kino XML/CSS processor. In *8th World Wide Web Conference*, Poster Session, Toronto, April 1999.

[12] E. Köppen, S. Nusser, and G. Neumann. Cineast – An extensible Web browser. In *Proceedings of the WebNet World Conference on WWW, Internet and Intranet*, November 1997.

[13] P. Leach and Y. Goland. WebDAV ACL protocol. *Internet Draft*, November 1997.

[14] L. Lippert. WebDAV access control goals. *IETF Internet Draft*, August 1998.

[15] E. Nebel and L. Masinter. Form-based file upload in HTML. *RFC 1867*, November 1995.

[16] G. Neumann and S. Nusser. Wafe – An X toolkit based frontend for application programs in various programming languages. In *Proceedings of the USENIX Winter 1993 Conference*, San Diego, California, January 1993.

[17] G. Neumann and U. Zdun. Filters as a language support for design patterns in object-oriented scripting languages. In *Proceedings of COOTS ‘99, 5th Conference on Object-Oriented Technologies and Systems*, San Diego, May 1999.

[18] J. Orton. Sitecopy. Web page, 1999. http://www.lyra.org/sitecopy/.

[19] D. Raggett, A. L. Hors, and I. Jacobs. HTML 4.0 specification. *World Wide Web Consortium Recommendation*, April 1998. http://w3c.org/TR/REC-html40/.

[20] K. Sikkel. A group-based authorization model for cooperative systems. In *Proceedings European Conference on Computer-Supported Cooperative Work (ECSCW’97)*, Lancaster, September 1997.

[21] G. Stein. mod_dav: A DAV module for Apache. Web page, 1999. http://www.webdav.org/mod_dav/.

[22] D. Wetherall and C. Lindblad. Extending Tcl for dynamic object-oriented programming. In *Proceedings of the Tcl/Tk Workshop 95*, Toronto, Ontario, July 1995.

[23] E. Whitehead and M. Wiggins. WebDAV: IETF standard for collaborative authoring on the web. *IEEE Internet Computing: Softwaere Enginering over the Internet*, 2(5):34–40, September-October 1998.

[24] W. Yoeng, T. Howes, and S. Kille. Lightweight directory access protocol. *RFC 1777, Standards Track*, March 1995.

[25] E. Young and T. Hudson. SSLeay. Web page, 1999. http://www.ssleay.org/.