

WebDAV based Open Source Collaborative Development Environment

Sunghun Kim, Kai Pan, E. James Whitehead, Jr.

*Dept. of Computer Science
University of California, Santa Cruz
Santa Cruz, CA 95064 USA
{hunkim, pankai, ejw}@cs.ucsc.edu*

Abstract

Open source projects are characterized by their distributed developers, openness, and use of a community-based development process. Successful open source projects such as Apache, Linux, and PHP have a web-based Collaborative Development Environment (CDE) that provides source code repository access, knowledge management, discussion lists, bug tracking, and user support. Distributed software development is also increasingly common for commercial software projects, and benefits from using web-based CDEs. GForge, derived from the SourceForge package, is an open web-based CDE that is seeing widespread open source and commercial use. To support remote authoring of web pages and file contents, as well as remote source code access, GForge uses several network protocols, including SSH/SFTP, CVS pserver, and FTP. These protocols inevitably introduce security risks and add administrative work. We modified GForge to eliminate use of FTP, SSH/SFTP, and CVS protocols, using instead a WebDAV-based integration. The new CDE, called Davenport, uses only the HTTP/WebDAV protocol, and HTTP/WebDAV replaces the other protocols without sacrificing functionality. In this paper we describe the architecture and features of Davenport, and show how WebDAV replaces existing network protocols. We discuss the future development directions of CDEs including integration of client side IDEs and using a CDE as a software test bed.

1. Introduction

Most successful open source projects are community-based, consisting of groups such as lead developers, committers, bug reporters, and users [1]. The community members work in a distributed environment, and have few, if any, face-to-face meetings. One key challenge of distributed development is communication among developers; one study reports that distributed work takes twice as long as collocated work due to communication issues [2].

To facilitate communication among the distributed developers in open source projects, various software development and collaborative tools are required. Software Configuration Management (SCM) tools such as CVS [3] are necessary to control source code, allow

changes to be made in isolation from other developers, and raise awareness of source code changes. To share ideas and store project knowledge, a discussion forum or mailing list is useful. A bug or issue tracking system is necessary to keep track of defects or new issues. Metrics such as download and page visit counters are helpful to estimate the number of users and popularity of the project. To advertise the project and to support users, a project homepage is essential.

Many open source project have a web-based, portal-style collaborative development environment (CDE) which integrates project tools. These web-based CDEs are a key part of active open source projects. For example, Apache [4] and PHP [5] projects have web-based CDEs which include CVS, a CVS viewer, a discussion forum, a mailing list, a bug tracking system, and a project homepage. The success of SourceForge [6], a web-based CDE hosting site, also highlights the importance of CDEs.

Commercial software development projects sometimes involve globally distributed teams, and can benefit from web-based CDEs. Several SourceForge-like CDEs have been developed to support distributed commercial projects, such as Source Cast [7], SourceForge Enterprise edition [8, 9], and GForge [10].

GForge provides a range of typical CDE services, including a project homepage, discussion forum, mailing list, user access control, bug tracking, a file release system, and CVS [10]. GForge is widely used in open source communities and software development companies including the Helix Community, Korean Linux document project, Philips, Thompson Financial, and NASA Goddard Space Flight Center [10].

To support CVS, homepage authoring, and the file release system, GForge makes use of several network protocols such as LDAP, FTP, SSH/SFTP, CVS pserver as well as HTTP. It raises security risks and administrative burdens.

The use of multiple protocols means that GForge must have multiple protocol servers running simultaneously. Each server is a separate code base, with its potential for security exploits. Hence, the more servers used by a CDE, the higher the chance for security holes. User login via SSH is especially risky due to the ability to have direct file system access, and the risk of users elevating their privilege illegally after system login [11].

Running several protocol handlers and servers requires additional administrative effort. An administrator must check if all servers are working well and ensure the services are running normally. If there are two or more data repositories that must be synchronized, e.g. DBMS and LDAP, it increases administrative work significantly. When these repositories are out of sync or fail to work, the administrator has to manually synchronize them. Additionally, the administrator must backup all repositories, and making consistent backups is a challenge across multiple repositories.

Collecting metrics requires more effort in a multi-server environment. For example the log files for SSH/SFTP, FTP, CVS, and HTTP are in different formats and in different locations. As a result, gathering metrics from all protocol handlers or servers requires a special log parser and analyzer for each of the log formats.

It would be beneficial to have all remote accesses to the CDE performed over one protocol with one administration interface, while eliminating the system login privileges to access the CDE server. Luckily such a solution exists today: the WebDAV protocol suite [12].

WebDAV is an extension to HTTP that enables distributed authoring and versioning [12]. WebDAV can replace FTP, SSH/SFTP, and CVS protocols, allowing CDEs to support their normal remote authoring and source code access functions using only HTTP/WebDAV.

WebDAV provides a superior webpage authoring mechanism as compared to direct file editing using SSH and a text editor, and eliminates the need for shell access to the CDE. Several well-known web content editors, including GoLive and DreamWeaver, already support WebDAV, and various WebDAV clients are available. File transfer using HTTP/WebDAV is trivial, and all major operating systems have native WebDAV drag and drop file upload/download capability. Subversion, a WebDAV based SCM, can be used as the SCM repository without adding a new server by configuring the Apache web server [13]. The unified and congregated HTTP log for homepage access, web authoring, file transfer, SCM usage, and CDE usage enables a CDE to provide abundant and detailed data to calculate metrics.

In this paper, we describe the architecture and features of Davenport, our WebDAV/Subversion-enabled version of GForge. We show how WebDAV can replace other protocols used in GForge without sacrificing existing functionality.

2. Rationale of WebDAV Unification

In this section we discuss the rationale of using WebDAV on top of GForge.

2.1. GForge

GForge uses a database to store most project information, including users, forums, bug tracking, and document data. The web interface is written in PHP [5], and the PHP functions interact with the database, Apache web server, and LDAP server. The simplified GForge architecture is depicted in figure 1.

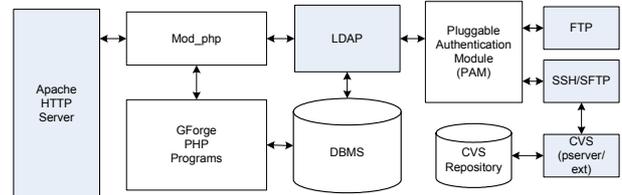


Figure 1. GForge Architecture. Protocol handlers are shown in gray.

To support homepage authoring and the file release system, GForge supports FTP and SSH/SFTP protocols. These protocols use a programmable authentication module with LDAP so that information authentication is performed by checking LDAP database instead of the `/etc/passwd` file for valid user id and password.

The CVS repository can be accessed through SSH tunneling using the same user id and password as for straight SSH, and CVS pserver is used to allow anonymous CVS access. In the backend, there are several UNIX crontab jobs synchronizing the database and LDAP, creating new CVS modules, creating group directories, gathering statistics data, etc.

As we have shown above, GForge uses several protocols and servers to provide its services. If there is a protocol that can handle all the capabilities provided by HTTP, SSH/SFTP, FTP, and CVS in GForge, it would reduce administrative work and security risks. WebDAV is such a protocol. We introduce WebDAV and compare it with other protocols in the next section.

2.2. WebDAV

WebDAV is a suite of protocol extensions to HTTP/1.1 [14] that supports collaborative authoring, namespace management of resources, setting and retrieving of metadata properties, access control, and resource versioning on remote web servers [12, 14]. The suite includes the DeltaV (versioning) [15], DASL (searching) [16], ACL (access control) [17], and WebDAV core (authoring) [12] protocols.

The WebDAV core protocol (RFC2518) enables users to create, modify, and delete resources or collections on a WebDAV server. WebDAV's concurrency control mechanism, whole resource locking, prevents overwrite conflicts during authoring. DeltaV is a versioning and configuration management protocol, and Subversion [13] uses WebDAV/DeltaV as a transport protocol. For a CDE like GForge, WebDAV can replace SSH/SFTP and FTP,

since it supports remote resource authoring functionality. CVS also can be replaced by Subversion, a version control system based on the WebDAV/DeltaV protocol [13]. The overview comparison between WebDAV and other protocols is shown in table 1.

Table 1. Protocol Feature Comparison

Feature	FTP	SSH	CVS	WebDAV
remote web authoring		✓		✓
remote file upload	✓			✓
versioning			✓	✓
Concurrency control (versioning)			✓	✓
concurrency control (web authoring)				✓
Secure connection		✓		✓

2.3. WebDAV vs SSH/SFTP/FTP

For homepage authoring and file transfer, GForge uses SSH/SFTP and FTP protocols. This practice is susceptible to security attacks. For example, several years ago, Apache Software Foundation and SourceForge servers were compromised due to a weakness in the SSH daemon [11]. Once an intruder achieves shell access, it is possible to illegally elevate privileges using the weaknesses of operating systems.

Let's consider a scenario, in which developers want to edit some project HTML files on a GForge server. With the SSH support in GForge, developers login to the server, and use a text editor on the server to edit the HTML files. In this case, it is hard to use specialized HTML editors such as GoLive or DreamWeaver. An alternative way is to download HTML files onto each developer's local disk, edit them using a HTML editor, and then transfer the changed HTML files to the server using SFTP or FTP. This process requires inconvenient extra work, and it can also introduce the lost-update problem when several developers transfer their data simultaneously to the server.

When using WebDAV instead of SSH/SFTP and FTP, these problems go away. WebDAV has a well-defined concurrency control mechanism: lock and unlock. There are already many WebDAV-compliant HTML editors such as DreamWeaver, GoLive and Adobe Photoshop. Windows, Linux, and Mac OS support WebDAV at the filesystem level, so developers can even mount a server-side Davenport homepage area as a local disk.

Overall, WebDAV provides superior homepage authoring, and it can replace SSH/SFTP and FTP in GForge, so that the administrative tasks and security risks are reduced.

2.4. Subversion vs CVS

The Subversion project was started to improve the functionality and fix the shortcomings of CVS. Use of Subversion confers many benefits as compared to CVS, including directory versioning, better concurrency control, and property support [13]. Subversion also addresses a long standing CVS issue: resource renaming in CVS causes loss of change histories. That is, when we change a resource's name or move a resource from one directory to another, CVS loses the version history of the resource, while Subversion keeps all history of renamed or moved resources. Subversion also supports data import and export functionalities like those of a DBMS.

Subversion eliminates the need for a specific protocol for SCM, such as CVS pserver or SSH tunneling. Subversion uses the WebDAV/DeltaV protocol, so it is possible to reuse the existing HTTP server in Davenport.

3. Davenport

In this section, we introduce the architecture of our WebDAV integration, Davenport. The Davenport architecture is shown in Figure 2. The additional features are also described.

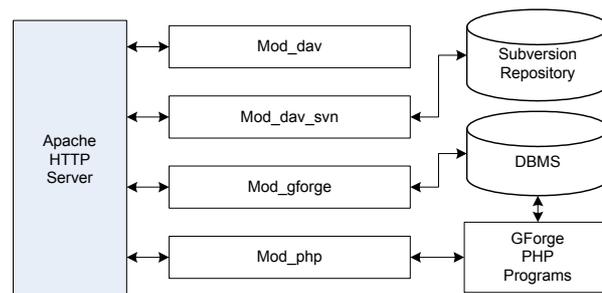


Figure 2. Davenport Architecture. The protocol handler, (Apache HTTP server), is shown in grey.

3.1. Architecture

We added the WebDAV integration using mod_dav [18] so that developers can author the project homepage using the WebDAV protocol rather than using SSH/SFTP or FTP. We also added a Subversion integration using mod_dav_svn [13].

For homepage authoring or source code check-in, a suitable permission is required. For example, only the member of a group should be able to change the source code of the group. To support HTTP authentication for WebDAV and Subversion, we wrote an Apache authentication module, mod_gforge, which retrieves user id/password information from a web browser, and compares it with the id/password record in the database to verify the user authentication.

Mod_dav, mod_dav_svn, and mod_gforge enable Davenport to replace SSH/SFTP, FTP, CVS and LDAP. Davenport uses the Apache HTTPD server for the web service, WebDAV, and Subversion.

Though adding new features and a new module, `mod_gforge`, may introduce new security risks, we believe this approach introduces much less risk compared to using multiple servers. Using WebDAV/Subversion integration improves GForge in several ways. First it reduces the number of different protocols supported; Davenport only needs to open ports for HTTP(S), which reduces administrative work and security risks. Second, Davenport reuses HTTP and its related technology such as basic authentication and SSL transport encryption. Third, the unified and aggregated HTTP log enables Davenport to provide detailed metrics. Finally, WebDAV and Subversion provide a superior authoring and version control environment.

3.2. Additional Features

In addition to the WebDAV/Subversion integration, we also added several new features, including SCM metrics, Subversion import/export, and Wiki discussions.

Gathering metrics about the project status and process is one of the major functionalities of SCM. For every project in Davenport, Subversion serves as a repository that records the status of all the components and SCM activities of the developers. Davenport collects project metrics, including commit numbers by each developer, the sum of lines changed by each developer, the commit number of each file, and the lines changed for each file, for a given period. With the help of these SCM metrics, displayed as graphs, the project members can supervise, analyze, and manage the development process, view the project metrics over time, and gather insight into each developer's work and status. The developers can also be aware of his/her peer developers' activities, as well as the whole status of the project.

One of the most frequent requests on SourceForge [6] is to import an existing CVS repository into SourceForge and to delete a specific module in CVS. Subversion handles this latter problem, and Davenport provides a web-based interface for Subversion's repository import function. A project administrator can upload a Subversion dump file and Davenport will automatically load the file into the Davenport Subversion repository.

Wiki is useful for discussion and scratch areas. Unlike mailing lists or discussion forums, Wiki provides simple and easy collaborative editing functionality through a web interface without using a special client. A good use of Wiki is sharing ideas or authoring frequently changing documents, such as a FAQ or TODO list. Davenport provides a separate Wiki space for each project using the virtual Wiki functionality in MoniWiki [19].

3.3. Administrative Improvement

Davenport only uses the HTTP/WebDAV protocol, and the administrator needs to monitor an `httpd` server.

There is no need to configure other servers such as LDAP or CVS. If there is need to restart servers, the administrator just needs to stop and start the `httpd` server. There is no need to synchronize DBMS with any other data repositories.

4. Current Experience

We are hosting 15 class and lab projects with 40 registered users using Davenport at University of California, Santa Cruz [20]. For class projects, students are required to write design and requirement documents, as well as UI prototypes, test cases, and their project source code. Usually a project group consists of three or four students. They use the Wiki to write project documentation, author their project homepage using WebDAV, and use document management to upload and store their documents. They use Subversion daily to store source code and keep all changes. An instructor also creates a group for the whole class, and uses it to share information about lectures and projects using Wiki pages, discussion forums and news. There is no need or requirement to support SSH/SFTP, FTP, or CVS access, and without those protocol supports, Davenport satisfies all needs for developers as a CDE.

5. Future Directions for Davenport

IDE integration is a high-priority planned activity of Davenport. So far, Davenport only provides Web interfaces for users to access most of its functionalities. This is not very convenient for developers who spend their time working in client-side IDEs like Eclipse. In our future work, we wish to build a seamless integration between Eclipse and Davenport. Here are several steps we will take.

First, a WebDAV client will be integrated into Eclipse to access the resources in the WebDAV repository in Davenport. Second, a Subversion client will be integrated into Eclipse to access the resources in the Subversion repository in Davenport. Third, we will build SOAP services into Davenport for most of its functionalities, such as mailing lists, news, task managers, bug tracking, Subversion metrics graphs, diaries and notes. Accordingly, SOAP client libraries will be developed for these Davenport SOAP services and these libraries will be integrated into Eclipse.

If the users write requirements documents or design documents with Wiki in Davenport, some mechanism should be built to allow the user to browse both the structure and contents of these documents in Eclipse. In this way, the users can read the requirements documents or design documents conveniently while coding, and document-to-code relationships can be established and saved.

To evaluate software tools and development process, empirical measurement and feedback are very useful. In that sense, CDEs can serve as a software engineering research test bed. If a research group develops a new tool, to improve the software development process, how can we validate this tool? Also if a research group develops a new software development process, how do we know the new process is better than others? The new tool or new process can in some cases be integrated into a CDE, and the CDE can collect metrics on the process or tool. To support this, the CDE has to have a fully pluggable architecture so that authorized groups or developers can add their own tools or process control on top of the CDE. Also, the CDE must provide detailed awareness functionality.

6. Related Work

Tools and environments are basic aids for software development, and have existed since the early days of computer programming [21]. From UNIX shells to IDEs, development environments keep evolving. We discuss web-based CDEs in this section.

SourceForge Enterprise Edition [8] is a commercial web-based CDE package which is derived from SourceForge.net [6]. Based on the SourceForge package, it adds several features, such as calculating metrics and enhancing interoperability. Still, it uses several dedicated network protocols, such as CVS and SSH/SFTP, which can be replaced by WebDAV.

Source Cast from CollabNet [7] is another commercial web-based CDE. Instead of supporting SSH/SFTP, it uses CVS areas for homepage storage. Each project has a special CVS module called www, and developers can check web resources into the CVS module to author the project homepage. Since CollabNet supports several Subversion developers, it seems likely they will integrate Subversion into Source Cast in the near future.

7. Conclusions

We have described the architecture of GForge and Davenport and how WebDAV can replace other protocols without sacrificing existing functionalities. Such WebDAV-based CDEs are more secure, reduce administrative overload, and can get benefits from many other WebDAV-aware tools.

Since WebDAV supports a superior environment for webpage authoring and version control environment, and reduces security risks and administrative work without sacrificing existing CDE functionality, we believe web-based CDEs should use WebDAV instead of several other protocols. Furthermore, the WebDAV protocol suite including DASL (searching), ACL (access control), is able to provide searching and detailed access control functionality for WebDAV-based CDEs, thus providing a

more functional, standard-based platform for future extensibility.

8. Acknowledgement

We would like to thank the GForge developers. Especially we'd like to thank Jennifer Bevan and Mark Slater for their invaluable discussion and review of this paper. This project is supported by the National Science Foundation under Contract Number NSF CAREER CCR-0133991.

9. References

- [1] C. Gacek and B. Arief, "The Many Meanings of Open Source," *IEEE Software*, vol. 21, 2004, pp. 34-40.
- [2] J. D. Herbsleb and A. Mockus, "An Empirical Study of Speed and Communication in Globally-Distributed Software Development," *IEEE Transactions on Software Engineering*, vol. 29, no. 6, pp. 481-494, 2003.
- [3] B. Berliner, "CVS II: Parallelizing Software Development," Proceedings of Winter 1990 USENIX Conference, pp. 341-351, Washington, DC, 1990.
- [4] Apache Foundation, "The Apache HTTPD Server Project," <http://httpd.apache.org/>, 2003.
- [5] The PHP Group, "PHP: Hypertext Preprocessor," <http://www.php.net/>, 2004.
- [6] VA Software, "SourceForge.net Homepage," <http://sourceforge.net>, 2003.
- [7] "CollabNet - SourceCast," <http://www.collab.net/products/>, 2004.
- [8] VA Software, "SourceForge Enterprise Edition," <http://vasoftware.com/products/>, 2004.
- [9] L. Augustin, D. Bressler, and G. Smith, "Accelerating Software Development through Collaboration," Proceedings of 24th International Conference on Software Engineering, pp. 559 - 563, Orlando, Florida, 2002.
- [10] GForge, "GForge CDE: Collaborative Development Environment," <http://gforge.org>, 2003.
- [11] B. Behlendorf, "Apache Software Foundation Server compromised, rescued," <http://archives.neohapsis.com/archives/apache/2001/0012.html>, 2001.
- [12] Y. Golland, E. J. Whitehead, Jr., A. Faizi, S. Carter, and D. Jensen, "HTTP Extensions for Distributed Authoring -- WEBDAV," *Internet Proposed Standard Request for Comments (RFC) 2518*, 1999.
- [13] B. Behlendorf, C. M. Pilato, G. Stein, K. Fogel, K. Hancock, and B. Collins-Sussman, "Subversion Project Homepage," <http://subversion.tigris.org/>, 2003.
- [14] R. Fielding, J. Gettys, J. Mogul, H. F. Nielsen, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," *Internet Draft Standard Request for Comments (RFC) 2616*, 1999.
- [15] G. Clemm, J. Amsden, T. Ellison, C. Kaler, and J. Whitehead, "Versioning Extensions to WebDAV," *Internet Proposed Standard Request for Comments (RFC) 3253*, 2002.
- [16] A. Babich, J. Davis, R. Henderson, D. Lowry, S. Reddy, and S. Reddy, "DAV Searching and Locating," *Unpublished manuscript, draft-davis-dasl-protocol-00*, 2000.
- [17] G. Clemm, A. Hopkins, E. Sedlar, and J. Whitehead, "WebDAV Access Control Protocol," *Internet-Draft, work-in-progress, draft-ietf-webdav-acl-09*, 2002.
- [18] G. Stein, "mod_dav: a DAV module for Apache," http://www.webdav.org/mod_dav/, 2003.
- [19] "MoniWiki Project Homepage," <http://moniwiki.sourceforge.net>, 2004.
- [20] "DavForge UCSC Homepage," <http://dforge.cse.ucsc.edu>, 2004.
- [21] H. Ossher, W. Harrison, and P. Tarr, "Software Engineering Tools and Environments: A Roadmap," Proceedings of the Conference on The Future of Software Engineering, pp. 263 - 277, Limerick, Ireland, 2000.