Of late, “virtual organisations” and collaborative work have come to prominence, largely due to the advent of reliable high-speed networking technology. While distributed collaborative programming is popular in many contemporary developmental environments, problems are common. These include poor communication, inefficient resource organisation and lack of cultural flexibility. In order to obviate or alleviate these problems and streamline product delivery, programming teams may use code sharing technology such as the popular IBM ProFrame and Sun SparcWorks.

Most recently, application developers have warmed to CVS, or “Concurrent Versions System”. The open source community, especially, has adopted the tool in the collaborative development of such popular projects as the Mozilla web browser and the Apache web server (now used in some 14 million websites globally, according to Netcraft.com, and covered earlier by Fielding 1999). While CVS does satisfy many of the requirements for collaborative development tools proposed by the literature (such as code reusability, version control, standardisation of interface and function, and support for fluid information sharing), there are several other reasons for the popularity of this new tool, and these merit some discussion.

Developed in 1989, CVS is a comparatively old tool that remains current through persistent revision from the open source community. CVS offers users a network and platform-independent code management tool that keeps track of code portions and revisions in a tree-like directory structure. When a programmer wants to work on a particular code fragment, they log into the CVS server and “checkout” the code: CVS typically maintains separate directories for each programmer working on the project, however this policy is largely the purview of the project manager. Once the programmer has finished working on that section of code and is satisfied with its functionality and validity, they check the code back into the CVS tree. CVS analyses the code for textual content changes and, providing the source code is not in conflict with other code fragments in the central tree, merges the new code fragment with
As with many open source applications, CVS is free for both commercial and non-commercial use and considerable peer support is available from other users in the open source community.

Possibly the most peculiar feature of the CVS tool is its treatment of file concurrency. Conventionally, code management tools will "lock" code fragments when users extract them from the development tree. This is generally to prevent syntactical and semantic errors associated with multiple users altering the same section of code, in the same way that database records are commonly locked while being updated. CVS differs from this with two very unorthodox approaches. First, CVS does not prevent other users from working on the same piece of code: in this way, the application's overall development is not dependent upon any individual programmer. Second, when altered code is committed back into the development tree, CVS "automatically" notes those portions of code that have changed and can update change logs accordingly (and independently of the user).

With the rise in distributed work and the availability of high speed internet from home (e.g. Stern et al. 2004), fewer users need to be in the same place to work on a given project. This work approach extends to research paper development, a task that typically takes many revisions over a protracted period, but that still benefits from having access to older versions of particular documents. While document versioning is part of the workflow for many teams, structured document versioning can be elusive. This can result in numerous effects:

- Access to documents is granted via shared folders, containing data sets, analytical results and textual documents.
- Documents typically have various file names to indicate the name of the work, the date on which it was last modified, who last worked on the document and a variety of cryptic additional letters to indicate idiosyncratic outcomes.
- Documents may become lost or corrupted without any recourse to immediate prior versions.
- Only one person can work on a document at a time, without prior agreement to the contrary. This can protract the writing process.
• A shared folder itself may be difficult to access when away from a desktop, resulting in the need to make sure that work files are available over email (e.g. from co-authors).

CVS may alleviate some of these problems by affording a structured library of documents from which the authoring team can independently draw their work materials. A single library can give the authors the ability to immediately track and inspect the various collections of changes made to their papers over time. This in turn would simplify not only the development process but also the submission and review process (particularly if authors change institutions or working conditions).

Nevertheless, a number of issues may impede the technology’s adoption in commercial organisations. In particular, these issues include a perceived lack of ongoing and reliable technology support. While proponents laud the friendly but professional approach to project development in open source software communities, businesses may feel that this is an insufficient source of support. The fiduciary relationship which exists between a firm and a commercial software vendor compels that vendor to support their software product.

Alternatively, firms may already be committed to using a particular processing platform, and the adoption of a new platform is either incompatible with open source software, or would involve too steep a learning curve in this regard.

While CVS is not a substitute for build trees or quality control methods and does not address all problems associated with distributed code development, it remains a useful tool for distributed programmers. The continued growth in popularity of the tool, despite its unorthodox approach to concurrency (and hence version control), suggests that the collaborative support and code management services offered by CVS are of substantial benefit to the distributed programming community.

References


