

# Considerações sobre controle de grupos em repositórios CVS durante a evolução de uma ferramenta web

Daniel Cárnio Junqueira\*, Renata Pontin de Mattos Fortes

<sup>1</sup>Instituto de Ciências Matemáticas e de Computação  
Universidade de São Paulo  
Av. do Trabalhador São Carlense, 400 – Caixa Postal 668  
13560-970 São Carlos, SP

danielcj@grad.icmc.usp.br, renata@icmc.usp.br

**Abstract.** *To perform version control activities at Open Source projects, systems as CVS are often used. However, considerable extra efforts are made to ensure that users control has been made in such projects, and in many cases the desired control is not reached. VersionWeb is a tool that is being evolved to ease the versions control activities and to provide a better users control at Open Source projects development. This tool shall work as an extra layer to CVS.*

**Resumo.** *Para realizar as atividades de controle de versões nos projetos de Software Livre, sistemas como CVS são frequentemente utilizados. No entanto, grande esforço extra é necessário para se fazer o controle de usuários nesses projetos, e, muitas vezes, o controle desejado não é alcançado. VersionWeb é uma ferramenta que está sendo evoluída para facilitar as atividades de controle de versões e prover melhor controle de usuários no desenvolvimento de projetos de Software Livre, funcionando como uma camada extra do CVS.*

## 1. Introdução

Um software é composto por diversos artefatos, entre eles arquivos de código e de documentação. Durante seu desenvolvimento, um controle sobre as alterações que ocorrem nos arquivos de código e documentação se faz necessário. Gerenciamento de Configuração de Software (GCS) é a disciplina responsável por esse controle sobre as mudanças [Pressman, 2001, Sommerville, 2000].

Para realizar as atividades de gerenciamento de configuração de software, são utilizadas ferramentas de controle de configuração e de controle de versões. CVS é a ferramenta atualmente mais utilizada nos projetos de software livre, apesar de diversas falhas já terem sido detectadas [Reis, 2003].

As principais carências de CVS estão relacionadas com poucos recursos para gerenciamento de usuários e falta de suporte para operações com arquivos (renomar ou mover arquivos) [Berliner, 1990]. Para corrigir as dificuldades de CVS com gerenciamento de usuários, recursos do sistema operacional ou mecanismos de controle de arquivos e usuários precisam ser utilizados. Mesmo com a utilização destes recursos, muitas vezes o controle de permissões para usuários não é tão completo como se desejaria. Já as falhas relativas a arquivos muitas vezes não são corrigidas, e os usuários precisam utilizar recursos mais complicados quando precisam realizar essas operações.

---

\* Apoiado pela FAPESP.

*VersionWeb* é uma ferramenta que foi inicialmente desenvolvida para funcionar como uma interface *web* para o CVS. Neste artigo relatamos uma evolução que está sendo feita na ferramenta a fim de torná-la uma camada extra para o CVS, que suporte o controle de usuários e permita o controle de permissões de acesso, além de oferecer mecanismos que permitam aos usuários efetuarem operações sobre os arquivos que estiverem em repositórios CVS.

O restante deste artigo está organizado da seguinte forma: a seção 2 faz uma breve apresentação de como são utilizados os sistemas de controle de versões nos projetos de software livre. A seção 3 apresenta *VersionWeb* e as modificações que estão sendo feitas para torná-la uma aliada ao desenvolvimento de software livre. Por fim, a seção 4 apresenta as considerações finais.

## 2. Controle de Versões nos projetos de Software Livre

As ferramentas de controle de versões são essenciais para o desenvolvimento de software livre. Elas são responsáveis pelo armazenamento da base do código dos projetos [Reis, 2003]. Essas ferramentas permitem que todas as mudanças de cada desenvolvedor sejam registradas, e, caso necessário, que recuperações de versões específicas do código possam ser realizadas.

Os projetos de software livre geralmente utilizam sistemas de controle de versões que possuem repositórios de código centralizados. O sistema mais utilizado nestes projetos é o CVS [van der Hoek, 2000]. Por seguir um modelo de desenvolvimento colaborativo, o acesso de leitura aos arquivos costuma ser sempre permitido a todos os usuários. No entanto, geralmente algum controle é exigido para acesso de escrita nos repositórios de código – dessa forma, todos os usuários podem submeter sugestões para os responsáveis pelos módulos dos projetos, e estes, se julgarem conveniente, integram as modificações ao repositório de código.

A fim de permitir ao CVS que faça o controle de acesso aos repositórios, várias configurações adicionais são necessárias, e, muitas vezes, recursos do sistema operacional são utilizados. Além disso, operações como mover ou renomear arquivos não são suportadas diretamente pelo CVS, e, para serem realizadas, exigem a utilização de comandos administrativos do CVS ou outros métodos complicados.

A próxima seção apresenta uma ferramenta que está sendo desenvolvida para permitir o controle de grupos no CVS.

## 3. VersionWeb

*VersionWeb* é uma ferramenta que foi originalmente desenvolvida como uma interface *web* para o CVS [Moreira et al., 2002, Soares et al., 2000]. Está sendo evoluída para que funcione como uma camada extra para o CVS, que permita o gerenciamento de grupos de usuários com a utilização do mesmo.

Uma estrutura hierárquica dos usuários foi planejada, conforme mostra a figura 1. Como é possível ver, as seguintes categorias de usuários estão sendo criadas:

**Administrador geral:** responsável pela ferramenta, pela configuração dos repositórios, criação de projetos e de administradores de projeto. Também podem criar outros administradores gerais.

**Administrador de projeto:** responsável por cada projeto. Cada projeto tem apenas um administrador, mas este pode criar áreas dentro do projeto, e outros usuários po-

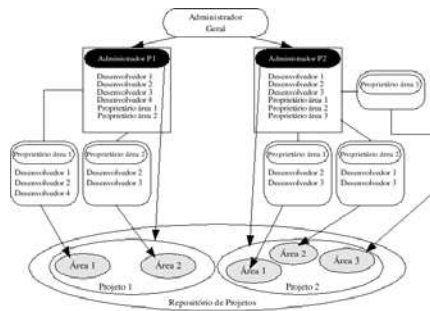


Figura 1: Estrutura hierárquica dos usuários em *VersionWeb*

dem ser os responsáveis por estas áreas. Isto pode ocorrer, por exemplo, quando se tem várias equipes, responsáveis por diferentes partes de um software.

**Proprietários de áreas:** são desenvolvedores que são denominados, pelo administrador do projeto, proprietários de áreas – no projeto Mozilla, por exemplo, são utilizados os “module owners” [Mockus et al., 2002].

**Autores:** são os desenvolvedores propriamente ditos. Os autores precisam ter permissão de escrita nas respectivas áreas em que estiverem trabalhando.

A ferramenta está sendo implementada utilizando linguagem Python. Sua arquitetura foi planejada conforme a figura 2: estão sendo implementados diferentes módulos, cada um responsável por uma parte da ferramenta. Com isso, os módulos poderão ser utilizados em outras ferramentas, sejam elas ferramentas que precisem apenas de funcionalidades de controle de versões ou das outras funcionalidades disponíveis nos módulos.

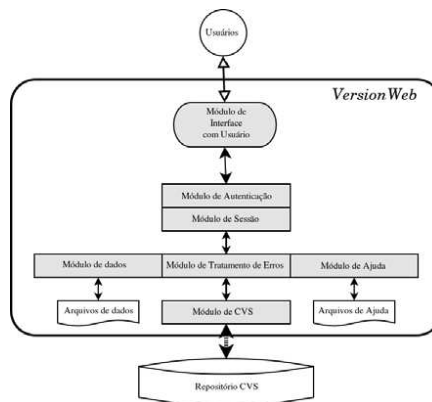


Figura 2: Arquitetura de *Versionweb*: utilização de módulos

Além de permitir o controle dos usuários no acesso a repositórios CVS, estão sendo incluídas rotinas em *VersionWeb* que devem permitir a realização de algumas operações sobre os arquivos, como mover e renomear arquivos, mantendo o histórico das versões. A ferramenta também está sendo integrada com *ViewCVS*<sup>1</sup> para permitir acesso público de leitura aos repositórios CVS.

<sup>1</sup>Disponível em <http://viewcvs.sourceforge.net>

#### 4. Considerações Finais

O desenvolvimento de software livre tem avançado de maneira extremamente rápida, e um acompanhamento na evolução das ferramentas de suporte se faz necessário. Nesse contexto, este artigo apresenta a evolução que tem sido realizada na ferramenta VersionWeb, que pretende oferecer melhor suporte ao desenvolvimento de software livre. Mais especificamente, a ferramenta em sua nova versão visa atender a demanda de trabalhos em grupos sobre os repositórios CVS. Neste artigo, foram apresentados os principais requisitos, relacionados ao suporte a trabalho em grupos, que estão sendo considerados.

Em seu estágio atual de desenvolvimento, a ferramenta funciona como uma camada extra ao CVS. No entanto, acreditamos que com apoio da comunidade em geral a ferramenta pode ainda evoluir muito, agregando inclusive funcionalidades do CVS e de outros sistemas de controle de versões, como o Arch<sup>2</sup>, que utiliza repositórios distribuídos, tornando-se assim uma alternativa aos sistemas atuais que implemente os mais novos conceitos de gerenciamento de configuração de software.

#### Referências

- Berliner, B. (1990). CVS II: Parallelizing software development. In *Proceedings of the USENIX Winter 1990 Technical Conference*, pages 341–352, Berkeley, CA. USENIX Association.
- Mockus, A., Fielding, R. T., and Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and mozilla. *ACM Trans. Softw. Eng. Methodol.*, 11(3):309–346.
- Moreira, D. d. A., Soares, M., and Fortes, R. P. d. M. (2002). VersionWeb: A tool for helping web page version control. *Linux Journal*.
- Pressman, R. S. (2001). *Software Engineering: A Practitioners Approach with E-source*. McGraw-Hill, 5th edition.
- Reis, C. R. (2003). Caracterização de um processo de software para projetos de software livre. Master's thesis, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação de São Carlos. Dissertação de Mestrado.
- Soares, M. D., de Mattos Fortes, R. P., and Moreira, D. A. (2000). Versionweb: a tool for helping web pages version control. In *International Conference on Internet Multimedia Systems and Applications*, pages 275–280, Las Vegas, EUA.
- Sommerville, I. (2000). *Software engineering*. Addison–Wesley, Wokingham [u.a.], 6th edition.
- van der Hoek, A. (2000). Configuration management and open source projects. In *Proceedings of the 3rd International Workshop on Software Engineering over the Internet*, Limerick, Ireland.

---

<sup>2</sup>Disponível em <http://www.gnuarch.org>