

La tortuga y los documentos: Tortoise + Subversion

Un sistema de control de versiones

Pablo Mellado Rafael Morales

Instituto de Astrofísica de Andalucía. CSIC

Seminarios IAA. Jueves 21 Junio 2007



Índice

- 1 Sistema de Control de Versiones (SCV)
- 2 Usos de un SCV
- 3 Subversion
- 4 Conceptos básicos
- 5 Problemas comunes
- 6 Conceptos avanzados



Índice

- 1** Sistema de Control de Versiones (SCV)
 - Definición de un SCV
 - Ejemplo de uso de un SCV Subversion
- 2 Usos de un SCV
- 3 Subversion
- 4 Conceptos básicos
- 5 Problemas comunes
- 6 Conceptos avanzados



Definición de un SCV

- Un "Sistema de Control de Versiones" (SCV) es una herramienta que permite gestionar las diferentes revisiones de un documento a lo largo de su ciclo de vida
- AKA: Revision Control ,Version Control, Source Control o (Source) Code Management (SCM) , Repository (repositorio)
- Se almacenan todos los cambios realizados en un fichero originando una nueva versión del mismo
- Es posible volver en cualquier momento a una versión anterior y detectar de forma automática todas las diferencias entre versiones
- Es posible reconstruir toda la historia de un documento desde su creación



Definición de un SCV (cont.)

- Trabajo concurrente entre usuarios
- Trabajo deslocalizado
- Control estricto sobre el manejo de versiones



Ejemplo de uso de un SCV Subversion

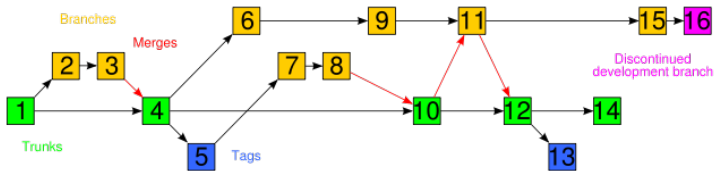


Figure: Ejemplo de uso de un repositorio Subversion

Índice

1 Sistema de Control de Versiones (SCV)

2 Usos de un SCV

- En proyectos
- En ficheros de texto
- En ficheros binarios

3 Subversion

4 Conceptos básicos

5 Problemas comunes

6 Conceptos avanzados



Usos de un SCV en proyectos

Uso recomendado en cualquier tipo de proyectos:

- multi/mono usuario
- código fuente
- artículo
- tesis
- página web
- proyecto de investigación/instrumentación



Usos de un SCV en ficheros de texto

Uso especialmente recomendado si se usan ficheros ASCII

- Código fuente de programas
- L^AT_EX
- XML
- HTML
- VHDL



Usos de un SCV en ficheros en binarios

Uso con ficheros propietarios / binarios

- Uso reducido de las características de un SCV
- Útil para guardar hitos



Índice

- 1 Sistema de Control de Versiones (SCV)
- 2 Usos de un SCV
- 3 Subversion**
 - Descripción
 - Subversion y CVS
 - Sistemas operativos soportados por Subversion
 - Listado de clientes Subversion
 - Tortoise, un cliente de Subversion
- 4 Conceptos básicos
- 5 Problemas comunes



Subversion, un SCV

- Web: <http://subversion.tigris.org/>
- Subversion es un Sistema de Control de Versiones de código abierto, gratuito que intenta ser un reemplazo de CVS



Subversion y CVS

- <http://subversion.tigris.org/>
- CVS es un sistema de control de versiones gratuito, de código abierto y ampliamente usado
- CVS y Subversion son equivalentes
- CVS sólo maneja ficheros de texto
- El almacenamiento de Subversion es más robusto al estar basado en una base de datos no en ficheros planos
- La gestión del almacenamiento en CVS es fácilmente modificable, la de Subversion no
- CVS está peor diseñado y documentado que Subversion
- Subversion está mejor integrado con la Web



Subversion y CVS (cont.)

- Subversion es más seguro en la autenticación y el cifrado
- Subversion dispone de transacciones atómicas, CVS no
- Subversion sólo transmite las diferencia entre versiones, CVS ficheros completos
- CVS ha sido ampliamente utilizado/revisado durante años, Subversion no



Sistemas operativos soportados por Subversion

- Solaris
- Linux
- Mac OS X
- Windows NT, 2000, XP y 2003
- IBM i5/OS (OS/400)



Listado de clientes Subversion

- Línea de comandos
- Interface Web
- Aplicaciones independientes
- Extensiones del 'shell' del Sistema Operativo (Tortoise)
- Plugging para IDE (Eclipse, JDeveloper)



Tortoise, un cliente de Subversion

- **Web:** <http://tortoisesvn.net/>
- Es un cliente gratuito de Subversion en código abierto y licencia GPL
- Implementado como una extensión de la 'shell' de Windows
- Implementa protocolos de autenticación y encriptación
- Integra un programa de gestión de diferencias
- Uso creciente (2 millones de descargas)
- Activamente desarrollado (bajo Subversion)

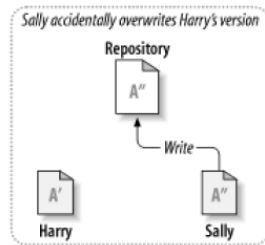
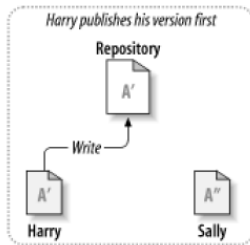
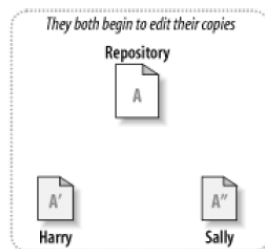
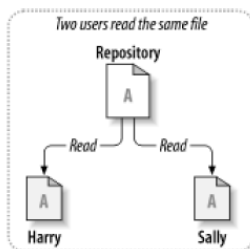


Índice

- 1 Sistema de Control de Versiones (SCV)
- 2 Usos de un SCV
- 3 Subversion
- 4 **Conceptos básicos**
 - El problema
 - Creación del repositorio
 - Importación de contenidos al repositorio
 - Vinculando localmente
 - Actualización de contenidos desde el repositorio
 - Actualización de contenidos hacia el repositorio
 - La Solución



El Problema a Evitar



Creación del Repositorio

- El repositorio se puede crear directamente en el servidor usando el comando: `svnadmin create`.
- También se puede crear a través del TortoiseSVN con la opción `Create repository here....`
- Un repositorio puede contener varios “proyectos”.
- Suponemos que tenemos instalado y funcionando un servidor `subversion` con su repositorio creado.



Importación de Contenidos al Repositorio (`import`)

- El concepto `import` tiene que ser visto desde el punto de vista del servidor para entenderlo mejor.
- El servidor importa los ficheros y directorios que le indiquemos.
- Para la importación de los ficheros usaremos la opción `Import`

Vinculando Localmente (`checkout`)

- Para gestionar los ficheros con `subversion` es necesario vincular una copia local al contenido del servidor.
- La opción `checkout` nos permite llevar a cabo esta tarea, y puede ser realizada por distintos usuarios.
- Es posible ejecutar `checkout` desde distintas máquinas con el mismo usuario.



Actualización Desde el Repositorio (`update`)

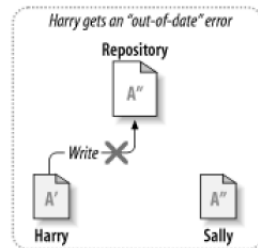
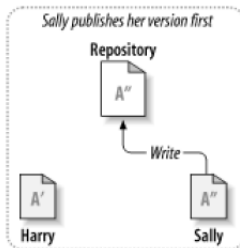
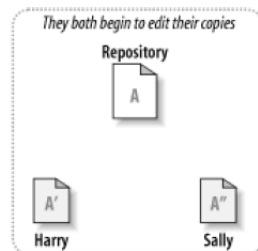
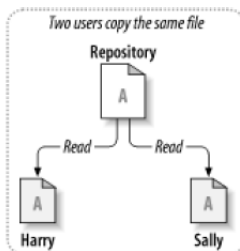
- La opción `Update` descarga del servidor los ficheros que han sido cambiados desde nuestra último acceso.
- Si hay nuevos ficheros o carpetas también serán descargados.
- Podría haber conflictos si al hacer `Update`, se descarga un fichero que ya había sido modificado localmente.



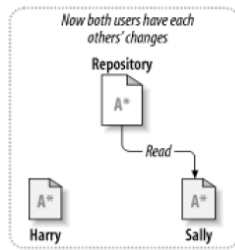
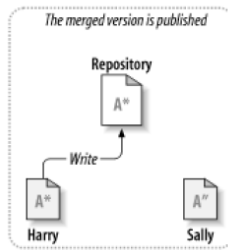
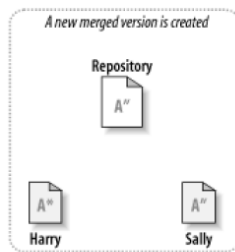
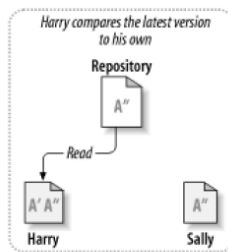
Actualización Hacia el Repositorio (`commit`)

- Es *muy aconsejable* ejecutar el comando `Update` antes de un `Commit`
- La opción `Commit...` envía las modificaciones locales al servidor.
- En cada `Commit` se incrementa el número de revision (`rev`).
- Si hay nuevos ficheros o carpetas aparecerá la opción de añadirlos al repositorio.

La Solución Aportada



La Solución Aportada (cont.)



Índice

- 1 Sistema de Control de Versiones (SCV)
- 2 Usos de un SCV
- 3 Subversion
- 4 Conceptos básicos
- 5 Problemas comunes
 - Resolución de conflictos entre versiones
 - Deshacer últimos cambios
- 6 Conceptos avanzados



Resolución de Conflictos entre Versiones (`resolve`)

- Cuando un conflicto aparece, se crean en el directorio local los siguientes ficheros:
 - `nombre_fichero.ext.rXX`, donde `XX` es la última revisión que "conocía" nuestro programa cliente
 - `nombre_fichero.ext.mine`, es el fichero tal y como lo teníamos antes del conflicto.
 - `nombre_fichero.ext.rYY`, donde `YY` es la nueva revisión que nuestro programa cliente "desconocía".
- En el fichero `nombre_fichero.ext` se hace una mezcla de todos los cambios de una forma bastante particular.
- Con la opción `Edit conflict` se puede resolver el conflicto gráficamente.
- Para dar el conflicto como resuelto usar la opción `Resolved`.



Deshacer cambios

Deshacer Últimos Cambios (`revert`, `diff`)

- Usando `revert` podemos deshacer todos los cambios y volver a la versión que nos descargamos del servidor.
- También es posible ver gráficamente estos cambios con `diff`.
- Usando la opción `show log` tenemos más opciones.



Índice

- 1 Sistema de Control de Versiones (SCV)
- 2 Usos de un SCV
- 3 Subversion
- 4 Conceptos básicos
- 5 Problemas comunes
- 6 Conceptos avanzados**
 - Etiquetas y Ramas
 - Fusionar diferentes versiones



Etiquetas (tag)

- Se utilizan para marcar determinadas revisiones.
- Se facilita la tarea si hemos utilizado la estructura recomendada (trunk, **tags**, branches).
- Muy usadas para "software release" o para "*congelar*" un estado.



Gestión de ramas (`branch`, `switch`)

- La creación de una rama es igual a la creación de una etiqueta, pero en este caso el directorio destino será `branches`.
- Muy usadas, por ejemplo, cuando se va a probar una nueva tecnología en nuestro desarrollo.
- Es necesario hacer `switch` si queremos que nuestros cambios se vayan realizando en la rama recién creada.



Fusionando diferentes Versiones (`merge`)

- Las ramas creadas pueden tener dos destinos finales:
 - 1 Ser abandonadas porque no interesa seguir su desarrollo. Para volver al trunk original es necesario hacer `switch`.
 - 2 Fusionarse con el `trunk` para incorporarlas al desarrollo principal. De nuevo es necesario hacer `switch` y despues `merge`.



Sumario

- Un SCV es una herramienta que permite gestionar las diferentes revisiones de un documento.
- Especial utilidad con ficheros ASCII pero también con ficheros binarios.
- Uso de un SCV recomendado en cualquier tipo de proyecto.
- Subversión es un SCV multiplataforma, multiusuario, gratuito y de código abierto. Tortoise es uno de los clientes de Subversion.



Sumario

- Crear un repositorio y realizar un `checkout` nos permite **gestionar versiones** de nuestros ficheros.
- Tenemos las herramientas necesarias para **resolver conflictos** y para volver a una **versión anterior**.
- Las **etiquetas** y las **ramas** nos permiten disfrutar de más funcionalidad.



Fin del Seminario

