UTILISATION DE CVS

Guillaume LEHMANN

Projet Rex

Version 2.3

Plan de la présentation (1/2)

Plan de la présentation (2/2)

Principes et définitions

1) Principes et définition (1/2)

La gestion de la configuration est une discipline qui permet d'identifier les composants d'un système en évolution continuelle, d'en contrôler les évolutions durant le cycle de vie du logiciel, d'archiver chacun des états successifs, et de vérifier que chacun de ces états est complet et cohérent.

Principes et définition (2/2)

- CVS est un logiciel GNU
- CVS est composé :
  - D'une partie cliente
  - D'une partie serveur
- Le repository est le répertoire racine du serveur CVS.
- Un module est un projet ou une partie d’un projet.

Fonctionnalités classiques

1) Principes et définition
2) Fonctionnalités classiques
3) Fonctionnement de CVS
4) Présentation des principales commandes
5) Politique de synchronisation
6) Conclusions
7) Pour aller plus loin
8) Copyright et Licence

2) Fonctionnalités classiques (1/2)

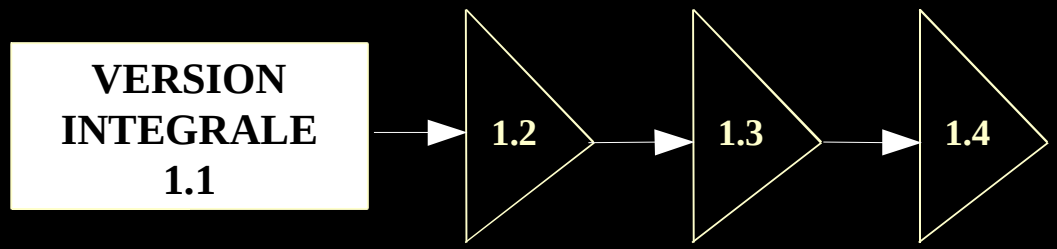
- Créer de nouvelles versions.
  - Ex : Versions “point-Oh”.
- Préserver les versions précédentes.
  - Ex : revenir à une version antérieure non-corrompue du fichier.
- Créer des variantes.
  - Ex : branches de développement, version stable et non-stable d’un fichier ou programme.
- Minimiser l'espace de stockage.
  - Ex : Seulement les deltas entres les versions des fichiers, par rapport au fichier final, sont conservés.

Fonctionnalités classiques (2/2)

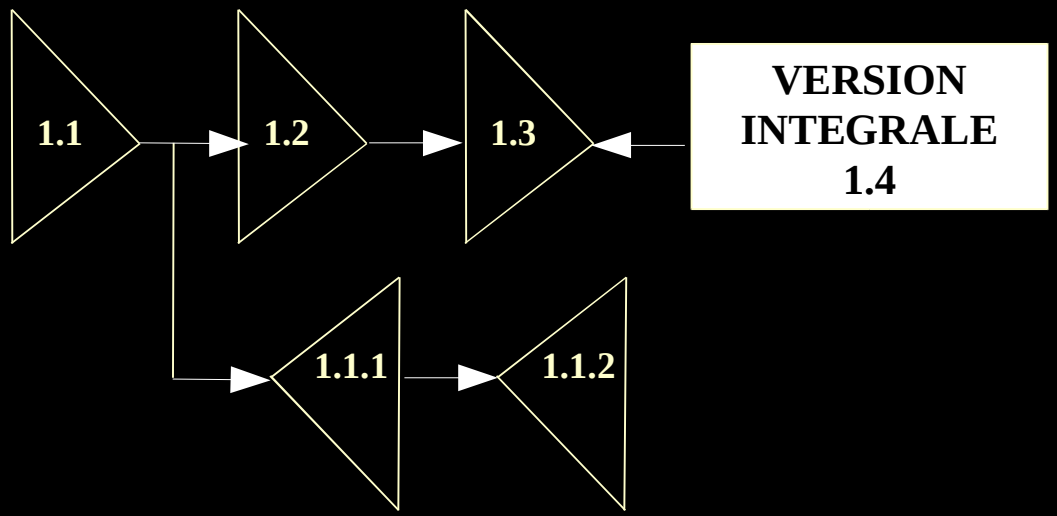
- Motiver les nouvelles versions.
  - Ex : Organiser ses objectifs par rapport aux versions point-oh.
- Maintenir la cohérence des unités entres elles.
  - Ex : Historique des modifications.
- Contrôler le partage concurrent de ressources.
  - Ex : éviter le travail redondant, et tous les problèmes que pose le travail d’une équipe sur un même fichier.
- Désigner logiquement des versions d'unités.
  - Ex : constitution d’un ensemble de fichier dépendant.

Fonctionnement de CVS

3.1: Mécanisme de stockage



Stockage par deltas avant (SCCS)



Stockage par deltas arrière (RCS, CVS)

3.2: Accès au dépôt CVS

- Mécanisme d’accès
  - Local,
  - Distant avec des protocoles RSH ou SSH,
  - Distant avec le protocole PSERVER.
- Mode d’accès
  - Mode client sur un module,
  - Mode serveur sur modules et base administrative.

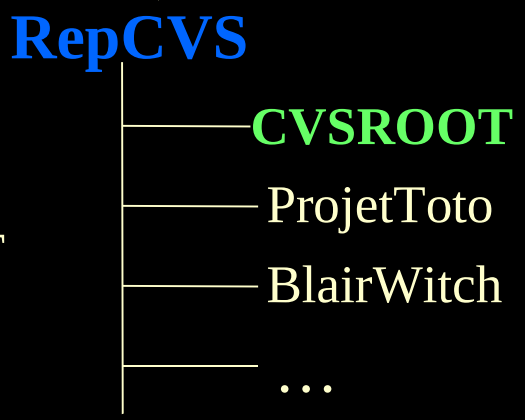
3.3: Créer un repository CVS et les modules

- **Un repository (ou dépôt) se compose de**
  - Modules contenant des projets,
  - Base administrative ou méta-module (CVSROOT).
- **Créer un repository CVS**

  #> mkdir $home/<cvs>

  #> cvs -d $home/<cvs> init

  #> cvs -d $home/<cvs> checkout CVSROOT



- **Créer un module**

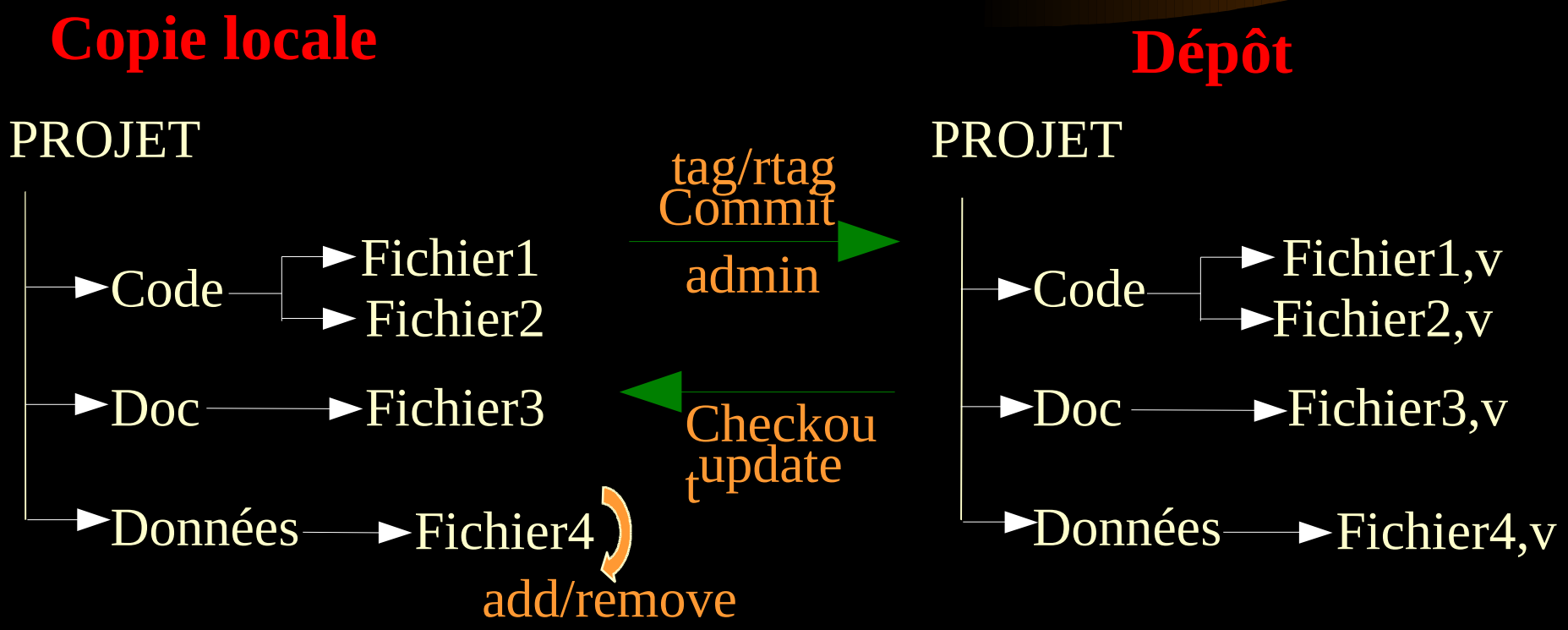
  #> cd <module>

  #> cvs -d $home/<cvs> import <module> <branche> <release>

  #> cd .. ; mv <module> <module>.old

  #> cvs -d $home/<cvs> checkout <module>

3.4: Copie locale / Repository



Présentation des principales commandes

...) ...

3) Fonctionnement de CVS

4) Présentation des principales commandes

- 4.1: Copie du fichier du serveur : checkout
- 4.2: Mise à jour de la copie locale : update
- 4.3: Mise à jour du repository : commit
- 4.4: Ajout d'un fichier au projet : add
- 4.5: Suppression d'un fichier au projet : remove
- 4.6: Numérotation des versions : tag et rtag
- 4.7: Verrouiller / Déverrouiller un fichier : admin -l / admin –u

5) Politique de synchronisation

...) ...

8) Copyright et Licence

4.1: Copie du fichier du serveur : checkout (1/3)

- **Copie à distance en pserver**
  - **Configuration du serveur :**
    - **Activation du cvspserver par inetd,**
    - **Fichier pour les mots de passe : $CVSROOT/CVSROOT/passwd et $CVSROOT/passwd :**
      - ***User :*** **utilisateur sur le client (différent du login qui suit)**
      - ***Password :*** **transmis en clair (ne pas utiliser celui de Linux)**
      - ***Login :*** **profil de l'utilisateur sur le serveur qui accède au repository. Si vide, Login = User.**

    Structure du fichier passwd : <User>:<Password crypté>:<Login>

Copie du fichier du serveur : checkout (2/3)

- **Copie à distance en pserver (suite)**
  - **Commandes clientes :**

#> cvs -d :pserver:<login>@<addrServ>:<$CVSROOT> login <module>

CVS password **********

#> cvs -d :pserver:<login>@<addrServ>:<$CVSROOT> checkout <module>

#> cvs -d :pserver:<login>@<addrServ>:<$CVSROOT> logout <module> (on peut se déloguer après plusieurs checkout /commit / update).

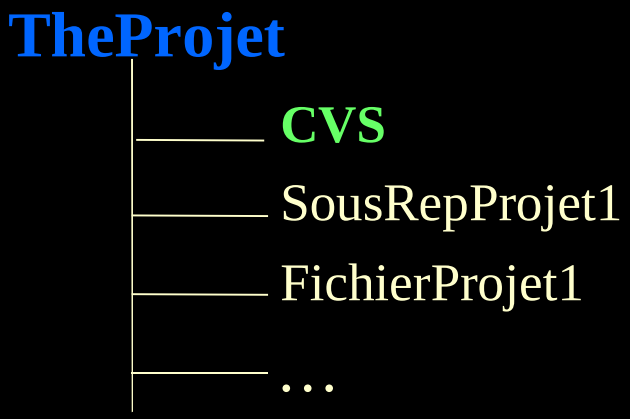
Copie en local du fichier : checkout (3/3)

- **Copie en local**

  #> cvs –d $CVSROOT checkout <module>

  #> cd <module>

- **Infos administratives dans le répertoire spécial CVS**



4.2: Mise à jour de la copie locale : update

- **Mise à jour d'un fichier :**

#> cvs update <NomFichier>

- **Mise à jour d'un projet :**

#> cvs update –P <module>

- **Résultat :**

U fichier1

C Doc

M fichier3

A	Fichier ajouté localement
U	Fichier ajouté par le dépôt (dépôt = repository)
P	Fichier mis à jour par le dépôt
M	Fichier modifié localement (éventuellement aussi par le dépôt mais sans collision)
C	Fichier modifié localement et par le dépôt avec collision

4.3: Mise à jour du repository : commit

- Mise à jour de la copie locale préalable pour éviter les collisions.
- Enregistrement des modifications apportées à la copie locale vers le dépôt :
  - Pour tous les fichiers dans l'arborescence,
  - Pour tous les fichiers dans le répertoire courant,
  - Pour un fichier spécifié sur la ligne de commande.
- Message décrivant les raisons de la modification obligatoire.

#> cvs commit –m <message> <NomfichierOuModule>

4.4: Ajout d'un fichier au projet : add

- Ajouter un fichier nécessite une mise à jour du dépôt.
- Distinguer un fichier binaire d’un fichier texte (un répertoire est traité comme un fichier texte) :

Fichier binaire : #> cvs add -kb <NomFichier>

Fichier texte : #> cvs add <NomFichier>

- Entériner l’ajout du fichier par un commit.

4.5: Suppression un fichier du projet : remove

- Retirer un fichier du projet nécessite une mise à jour du dépôt.
- Suppression :

#> cvs remove <NomfichierOuRep>

- Entériner la suppression du fichier par un commit.
- Le fichier est supprimé du projet, mais peut être retrouvé, car le serveur CVS le sauvegarde.

4.6: Numérotation des versions : tag et rtag

- Imposer le choix d'un numéro de version :

#/<module>/Code> cvs commit -r2.0 fichier2

- Donner un nom logique aux versions de fichiers
  - Retrouver rapidement un ensemble cohérent de fichiers,
  - « Geler » une version opérationnelle du projet.
- Etiquetter avec un nom logique
  - Un module (rtag),
  - Une arborescence de fichiers (tag).

#/<module>> cvs rtag version_stable-1_0 <module>

#/<module>> cvs tag version_stable-1_0 <fichiers>

4.7: Verrouiller/Déverrouiller un fichier: admin -l / admin -u

- Rapatrier le fichier en local et le verrouiller:

#> cvs checkout <module>

#> cvs admin –l <NomFichier>

- Un commit ne déverrouille pas un fichier.
- N’importe qui peut déverrouiller le fichier :

#> cvs admin –u <NomFichier>

Politique de synchronisation

5) Politique de synchronisation

- Mettre à jour fréquemment sa copie locale
  - Eviter les collisions,
  - Eviter de travailler décalé.
- Mettre à jour le dépôt sous certaines réserves
  - Enregistrer avec l'assurance de ne rien casser,
  - Vérifier que la mise à jour préalable de la copie locale ne remet pas en cause vos modifications,
  - Fournir un message justifiant la modification
    - Non pas comment elle a été faite,
    - Mais pourquoi elle a été faite.

Conclusions

6) Conclusions

- Ne pas hésiter à utiliser CVS pour :
  - Prémunir contre des erreurs d'inattention,
  - Prémunir contre les erreurs de disque,
  - Présenter un projet toujours en état,
  - Travailler en équipe,
  - Gagner du temps.
- Se familiariser avec un gestionnaire
  - Un plus dans l'industrie – Valorisant,
  - CVS est fortement utilisé dans les projets extérieurs.

Pour aller plus loin

1) Principes et définition
2) Fonctionnalités classiques
3) Fonctionnement de CVS
4) Présentation des principales commandes
5) Politique de synchronisation
6) Conclusions
7) Pour aller plus loin
8) Copyright et Licence

7) Pour aller plus loin (1/2)

- Commandes utiles :
  - log,
  - diff.
- Insertion automatique dans le document :
  - du numéro de version,
  - d'infos sur le document (auteur, date, ...).

Pour aller plus loin (2/2)

- http://www.cvshome.org/
  - Site officiel de cvs,
- http://www.wincvs.org/
  - Site officiel de wincvs,
- http://www.gnu.org/software/cvs/cvs.html
  - présentation de cvs pas gnu,
- http://www.gnu.org/manual/cvs/html_mono/cvs.html
  - Manuel complet sur cvs en anglais.

Copyright et Licence

1) Principes et définition
2) Fonctionnalités classiques
3) Fonctionnement de CVS
4) Présentation des principales commandes
5) Politique de synchronisation
6) Conclusions
7) Pour aller plus loin
8) Copyright et Licence

8) Copyright et Licence