CVS

Gestionnaire de versions concurrentes

08/01/2004

Didier Vojtisek

Plan

- Survol des fonctionnalités de CVS
- Utilisation
  - Gestion de versions
  - Gestion de la concurrence
- Utilisation avancée
  - Branches
  - Gestion de la base
  - Macro expansion
  - Alertes
- Conclusion

Outils existants

- ClearCase
- Continuus
- PVCS
- Visual SourceSafe
- **CVS : C**oncurrent **V**ersions **S**ystem
- …

3

CVS pourquoi faire ?

- **Travailler à plusieurs sur les mêmes fichiers en même temps (concurrence)**
- **Gérer les versions des sources d'un développement**
  - **Tags symboliques**
  - **Comparaisons entre versions**
  - **Multiple lignes de développement dans une seule base**
  - **Gestion des branches de développement**
  - **Support configurable pour les log**
  - **Support pour les fichiers binaires**
  - **Envois d'évènements**

4

Ce que CVS ne fait pas

- **Système de construction**
- **Suivi de bugs**
- **Suivi de dépendances**
- **Procédure de test**
- **Gestion de la documentation du code**

5

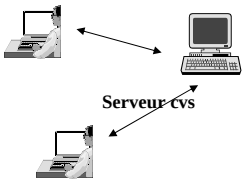
Disponibilité

- Logiciel libre disponible pour toute plate-forme
- Client en ligne de commande
- Plusieurs clients graphiques
  - **WinCVS, tkCVS,** jCVS, WebCvs, Jalindi igloo, TortoiseCVS …
- **Clients intégrés à des outils**
  - **Eclipse, …**
- **Un serveur dédié à l ’Irisa (sur le modèle de celui de Sofia) (en fait ssh)**

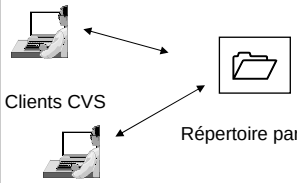
6

Architecture

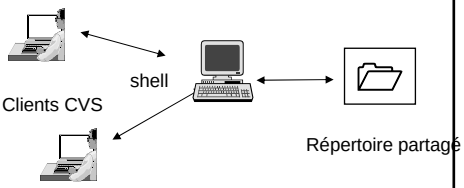
- 3 modes:



Client/serveur



Local

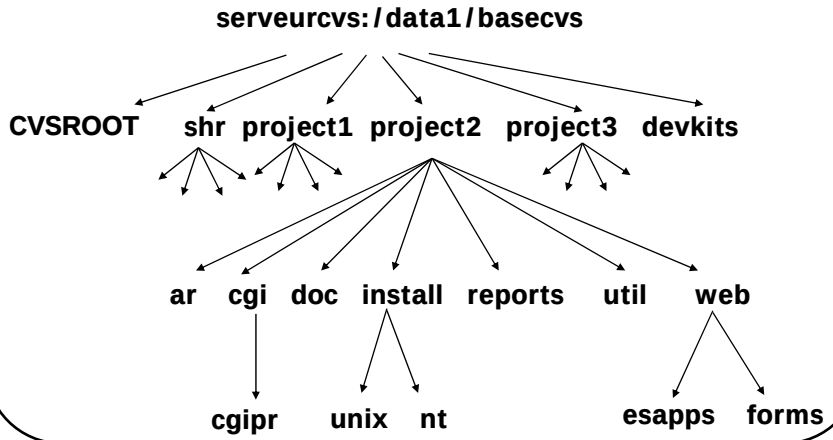


Distant

7

La base CVS

- **Réside sur un serveur**
- **Pas de fichier de travail dans la base**



8

Plan

- **Survol des fonctionnalités de CVS**
- **Utilisation**
  - **Gestion de versions**
  - **Gestion de la concurrence**
- **Utilisation avancée**
  - **Branches**
  - **Gestion de la base**
  - **Macro expansion**
  - **Alertes**
- **Conclusion**

Modèle d'utilisation

Checkout, Commit, Update

- Checkout

  Crée une copie privée dans un répertoire de travail. copies multiples, version multiples possibles.
- Commit

  Valide les changements d'une copie de travail pour les ajouter à la base. La copie de travail doit être à jour avec la base
- Update

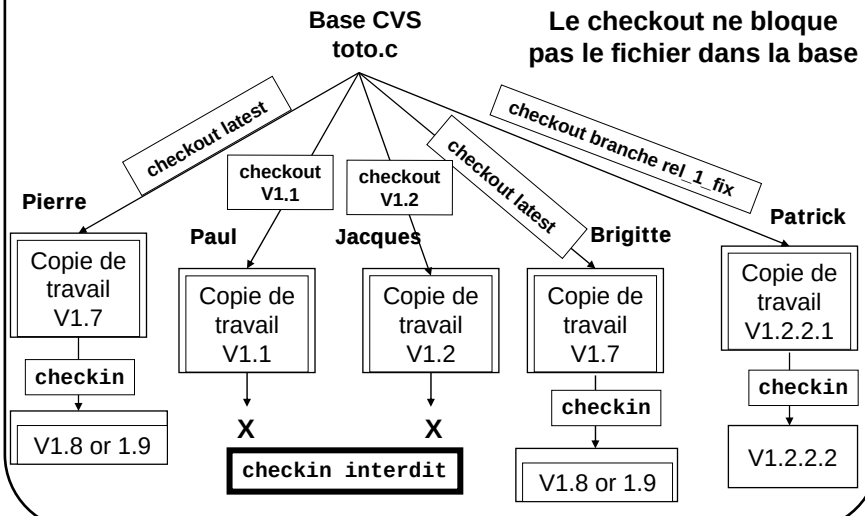
  Met à jour une copie de travail à partir de la version dans la base

Révision et tag

- **Révision**
  - **Numéro qui s'incrémente à chaque changement validé dans la base.**
- **Tag (étiquette)**
  - **Pour marquer des étapes où l'on souhaite pouvoir revenir ultérieurement**
  - **Pour marquer une version distribuée.**
- **Remarque :**
  - **Les numéros de révisions ne sont pas liés aux versions du logiciel.**

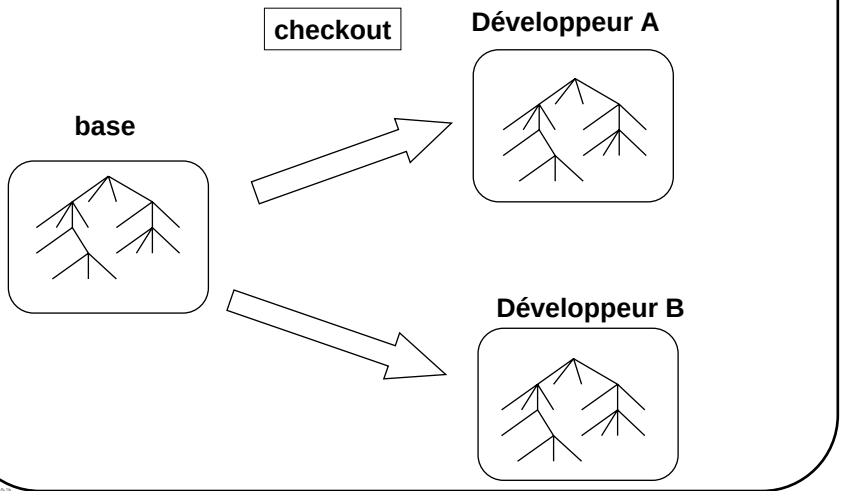
11

Checkout concurrents

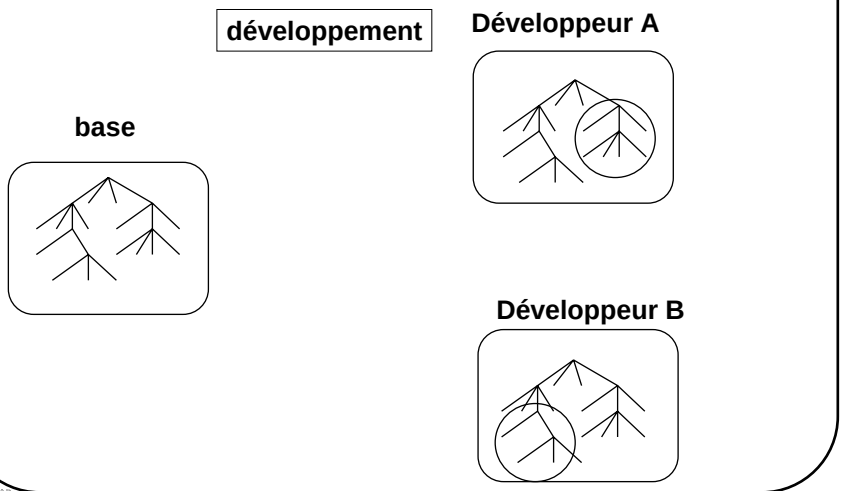


12

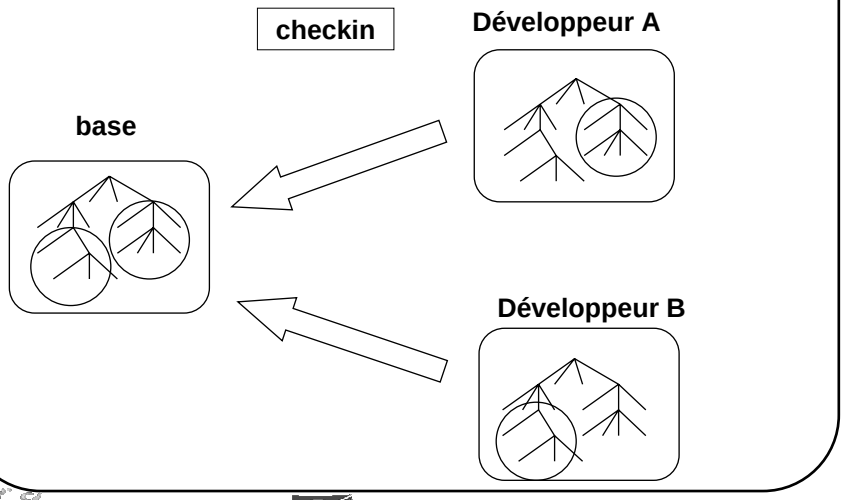
Développement idéal (1/4)



Développement idéal (2/4)

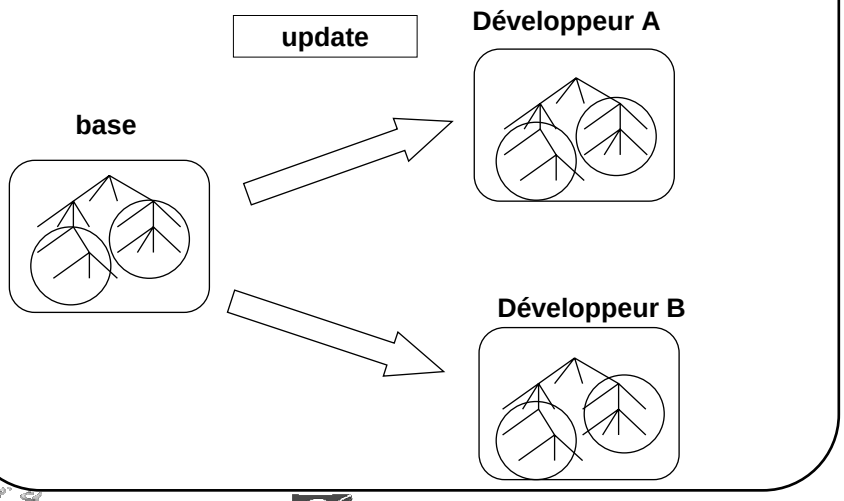


Développement idéal (3/4)



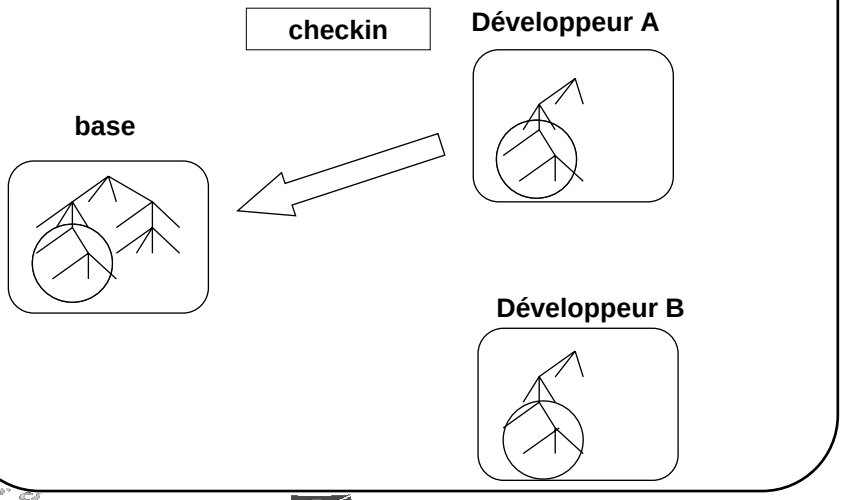
15

Développement idéal (4/4)



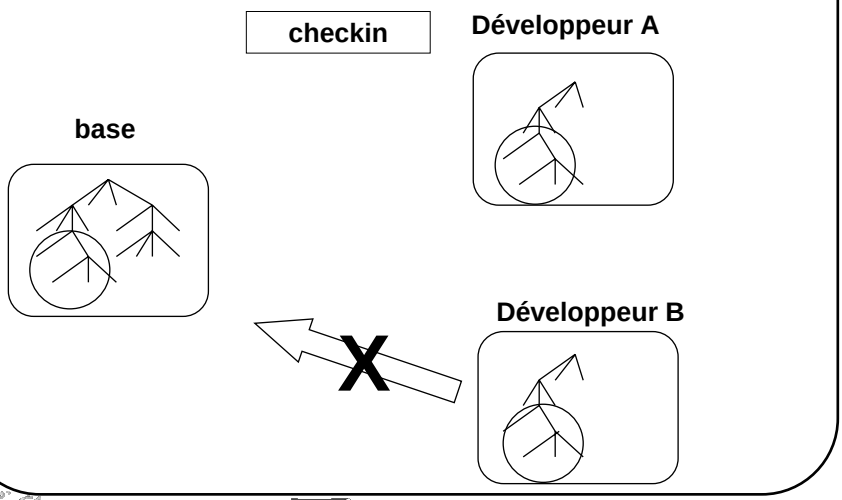
16

Développement réel (1/5)



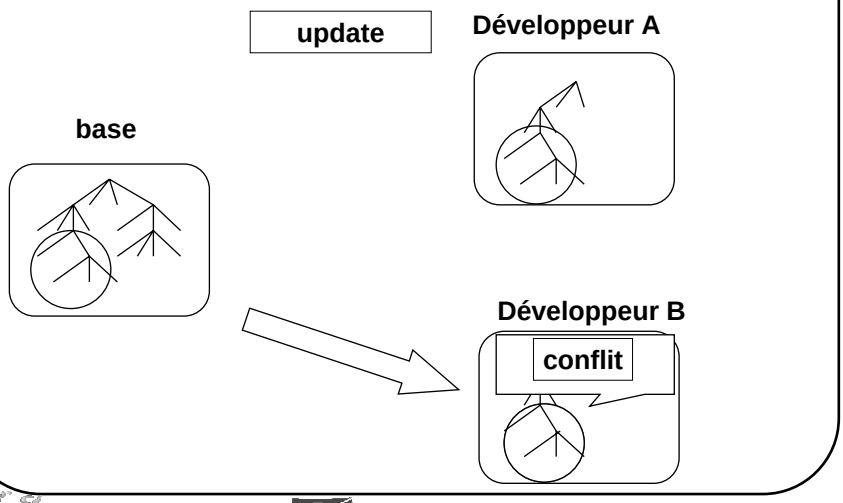
17

Développement réel (2/5)

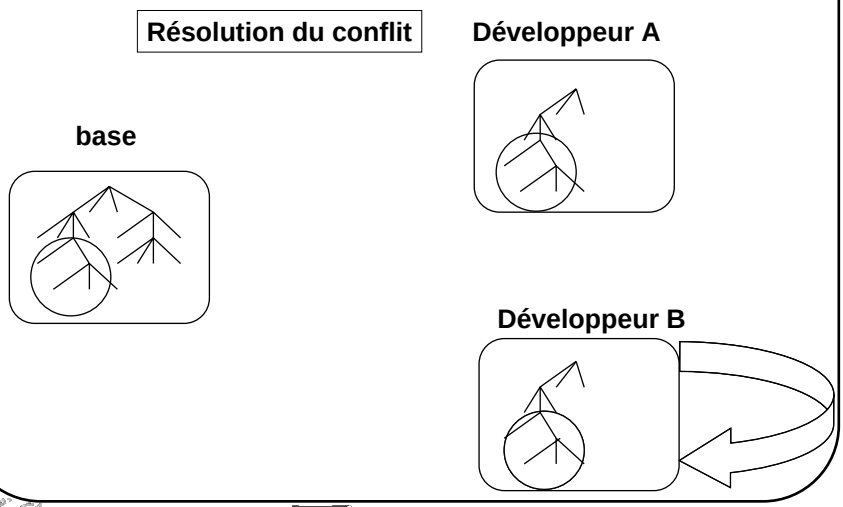


18

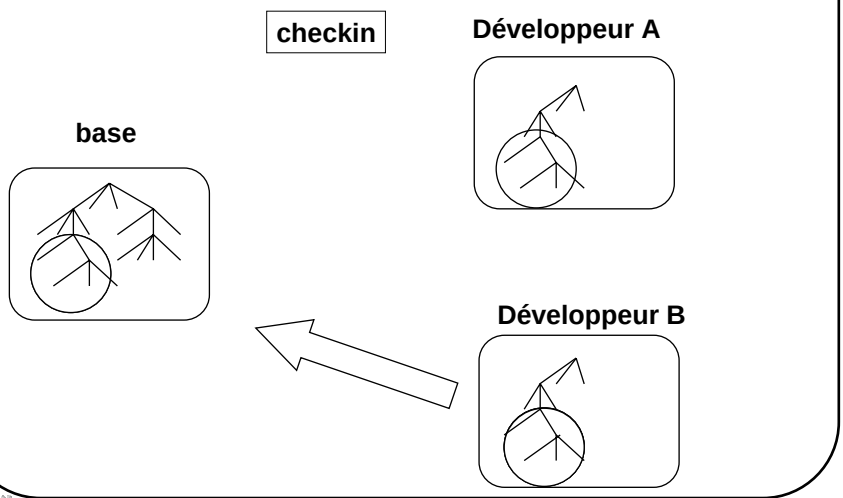
Développement réel (3/5)



Développement réel (4/5)



Développement réel (5/5)



Synchronisation avec update (1/2)

- (?) unknown : le fichier n 'est pas connu dans CVS. Utiliser cvs add.
- (A)dded : l 'utilisateur a ajouté un nouveau fichier mais il n 'est pas encore rendu public (commit)
- (U)pdated : le fichier n 'existait pas dans la copie de travail ou n 'avait pas été modifié localement. La nouvelle version remplace l 'ancienne.
- (R)emoved : Le fichier a été supprimé de la base, il est détruit de la copie locale.
- (M)odified : le fichier a été modifié localement, mais il n 'a pas encore été rendu public

Synchronisation avec update (2/2)

- **(M) erged : Le fichier a été modifié par un autre utilisateur mais aussi localement. Cependant CVS a réussi à effectuer une fusion sans conflit. Une copie de sauvegarde est créée dans la copie privée.**
- **(C) onflict : Le fichier a été modifié par un autre utilisateur mais aussi localement. CVS a détecté un conflit lors de la fusion. Une copie de sauvegarde est créée.**

Résolution d'un conflit

- **Commande " diff "**
- **Outils graphiques montrant les deux versions.**
- **Eviter les conflits :**
  - **penser à faire un update régulièrement**
  - **Chaque intervenant doit travailler sur des domaines fonctionnels séparés.**
  - **Communiquer**

Concurrence sur des fichiers non texte

- **Pas de diff disponible, les conflits sont à éviter à tout prix !**
- **Dans ce cas, CVS sert surtout pour l'historique.**
- **Une solution si les accès concurrents sont absolument nécessaires:**
  - **Utiliser une copie locale dans un répertoire d'équipe et laisser gérer le problème de concurrence par l'outil qui ouvre le fichier**
  - **Nécessite de bien informer les utilisateurs sur les zones qu'ils peuvent manipuler depuis leur copie locale et depuis la copie locale d'équipe.**

Commandes habituelles

cvs [*cvs-options*] *command* [*cmd-options*] [fichiers]

cvs init	Initialise une nouvelle base.
cvs checkout	Récupère des fichiers pour édition.
cvs commit	Valide les modifications vers la base
cvs update	Met à jour la copie locale avec la base.
cvs add	Ajoute des fichiers/répertoires dans la base.
cvs remove	Supprime un élément dans la base.
cvs tag	Labellise les sources.
cvs status	Montre le status des fichiers locaux.
cvs log	Affiche l'historique des révisions.
cvs diff	Compare les fichiers de la copie de travail avec les versions de la base.

Suppression / Renommage

- **Suppression**

  répertoire sur la base : Attic (grenier)

- **Renommage**

  Il n'existe pas de mécanisme spécifique

  => delete + add (discontinuité de l 'historique)

Modules

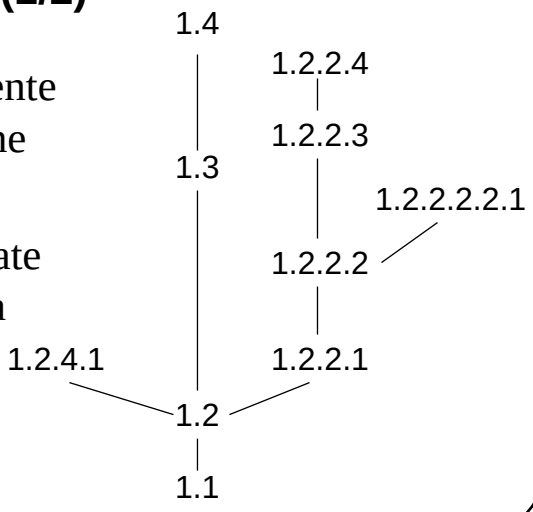
- **Arborescence de répertoires**
- **Parties visibles si l'on se connecte à distance sur un système inconnu**
- **Utilisé en particulier par certaines interfaces graphiques**
- **Composition de modules possible**
  - **Alias : traite plusieurs modules comme un seul module**
  - **Inclusion d'un module dans un autre**
  - **Exclusion**
  - **…**

Plan

- **Survol des fonctionnalités de CVS**
- **Utilisation**
  - **Gestion de versions**
  - **Gestion de la concurrence**
- **Utilisation avancée**
  - **Branches**
  - **Gestion de la base**
  - **Macro expansion**
  - **Alertes**
- **Conclusion**

Branches (1/2)

- correction urgente d'un bug sur une version livrée
- commit et update sans troubler la branche principale



Branches (2/2)

- Toujours planifier de joindre ou d'abandonner une branche (utilisation de la fusion de CVS)
- A utiliser avec précaution …
  - La jonction de branches pose pas mal de problèmes, en particulier lors de branches multiples

Gérer une base

- Choisir son emplacement
- Gérer les modules
- Définir les politiques d'utilisation
  - Actions spécifiques lors d'évènements (généralement un envoi de mail)
  - Watch/edit

Configurer une base

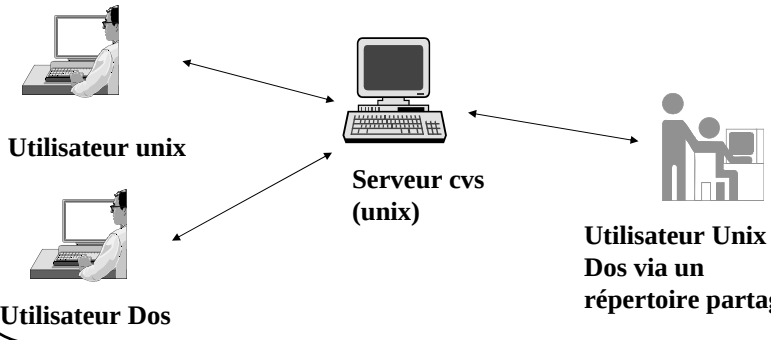
- **Module spécial CVSROOT**
  CVS checkout CVSROOT
- **Différents fichiers de configuration**
  - **modules, définitions des modules dans la base**
  - **cvswrapper aide à gérer le problème des fichiers binaires**
  - **cvsignore indique les fichiers que CVS doit ignorer (*.o, …)**
  - **commitinfo, editinfo, loginfo, notify, rcsinfo, taginfo, permettent de configurer des actions suivant les opérations effectuées**

Mode de stockage standard

- **Basé sur RCS**
- **Fichiers textes par défaut**
- **Utilisation de la commande diff**
- **Stockage des différences entre 2 versions successives.**

Problème Unix/Dos

- **Spécificité de l'Inria : travail en environnements multiples**



35

Unix/Dos

Cas Normal

Texte initial (Dos) :
Mon texte {CR}{NL}

Import avec un client Dos
Texte sur le serveur (Unix) :
Mon texte {NL}

Reconnaissance des fins de ligne

Export à partir d'une station
Texte obtenu (Unix) :
Mon texte {NL}

Expansion des fins de lignes en fonction du client

36

Exemple de problème Unix/Dos

Texte initial (Dos) :

Mon texte {CR}{NL}

Import à partir d'une station Unix

Texte sur le serveur (Unix) :

Mon texte {CR}{NL}

Export à partir d'un PC

Texte obtenu (???) :

Mon texte {CR}{CR}{NL}

Expansion des fins de lignes en fonction du client

Solution

- **Créer des copies de travail pour chaque type de système d'exploitation**
- **Toujours utiliser le client CVS du système qui va manipuler les fichiers.**

Macro expansion

- **Mots clés à insérer dans les sources**
  - **$Author$**
  - **$Date$**
  - **$Revision$**
  - **Id**
  - **log**
  - **...**

Exemple de macro expansion

pour un script shell ou un Makefile.

```
# $Id: makefile,v 1.2 2002/01/23 09:21:33 auser Exp $

# $Log: makefile,v $
# Revision 1.2 2002/01/23 09:21:33 auser
# Correction d'une typo dans le makefile.
#
# Revision 1.1 2002/01/01 10:55:30 anotheruser
# Version initiale
#
```

Attention aux macro expansion

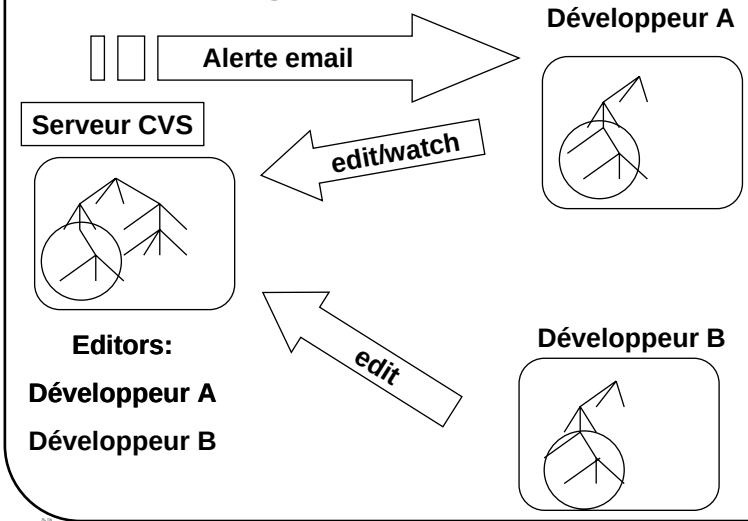
- **À utiliser généralement dans une partie en « commentaire »**
- **Log copie le début de ligne en supposant que c'est la mise en commentaire**
- **Attention aux fichiers dont le format est inconnu**

- **Solution:**
  - **désactiver l'expansion –ko**
  - **le passer en tant que fichier binaire -kb**

Edit/Watch

- **Ajout au fonctionnement standard**
  - **Les fichiers sont récupérés en lecture seule. Il faut utiliser la commande edit pour faire des modifications**
  - **on peut configurer CVS pour que chaque modification envoie un mail**
    - **pratique pour les parties de projet évoluant peu.**
    - **Etant donné que la notification passe par mail, cela peut être gênant pour un gros projet.**

Advisory locks



Création de la base

- **Import**
  - **Attention les fichiers qui ont servi à initialiser un module ne sont pas connecté à la base CVS !**
  - **=> Procédure**
    - **Import des fichiers**
    - **Renommage des originaux**
    - **Checkout**
    - **Supprimer les originaux (après vérification)**
    - **Pour enfin travailler …**

Choix du mode de connexion

Mode		Utilisation
mode local (accès fichier)		locale à l'Irisa !!! Problèmes potentiels sous win32
mode distant	par remote shell	locale à l'Irisa
	par remote shell sécurisé	de l'extérieur vers l'Irisa
	par remote shell sécurisé sur la machine lievre	de l'extérieur vers l'Irisa, sans compte à l'Irisa
mode serveur		pas d'utilisation à l'Irisa

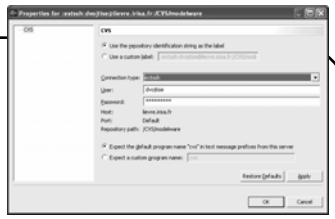
- **Mode distant recommandé**
  - **Autorise l'envoi de mail**
  - **rsh peu contraignant**

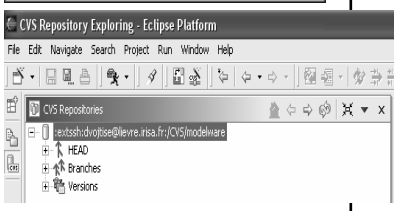
rsh sous windows

- **Simplement ajouter votre machine dans votre .rhosts**
- **Préférer le rsh de cygwin**

CVS et Eclipse

- Un vue supplémentaire (gestion des serveurs et fenêtre de synchronisation), des menus contextuels
- Methode de connection :
  - ext pour rsh ou ssh
  - extssh pour ssh mais sauvegarde du mot de passe dans eclipse





Derniers conseils

- **CVS n ’est qu’un outil de gestion de versions, il ne dispense pas d ’un bonne organisation du projet et du développement**
- **Ne pas attendre trop longtemps pour se resynchroniser**
- **quelques idées :**
  - **noter la version des outils utilisés (historique ou fichier spécial)**
  - **avoir en permanence une version extraite en lecture seule (éventuellement compilée) utile pour la consultation des documentations, tests de non régression, ...**

Aller plus loin

- **Page de l'irisa**
  - **http://www-interne.irisa.fr/atelier/documentations/ASCII/logiciels/CVS.html**
- **Général**
  - **www.cvshome.org**
  - **www.loria.fr/~molli/cvs-index.html**
- **GUI**
  - **www.twobarleycorns.net/tkcvs.html**
  - **www.wincvs.org/**
  - **www.jalindi.com/igloo**
  - **www.jcvs.org**

Questions/réponses