

Projet de développement

Philippe Collet

Licence 3 Informatique

2009-2010



Objectifs

- ❑ Réalisation, en équipe,
 - d'un développement logiciel de taille conséquente
 - à partir d'un cahier des charges et d'une architecture préétablis

- ❑ Donc : du développement !
 - Sans (gros) problème de conception
 - Avec des problèmes de
 - ◆ Communication (équipe de 4)
 - ◆ Techniques de programmation (API, etc.)
 - ◆ Fiabilité (tests unitaires indispensables)

- ❑ Travail demandé
 - **Beaucoup, beaucoup (beaucoup !) de travail personnel**
 - Surtout par rapport au faible volume des TD



Calendrier et évaluation

❑ Calendrier

- vendredi 12 mars 2010 : date limite de retour par mail des choix des étudiants, groupé par équipe
- jeudi 18 mars 2010 : publication des affectations
- jeudi 25 mars 2010 : 1er TD de suivi
- ⚠️ Merci aux étudiants participants à la ronde des facs de s'assurer qu'au moins un membre de leur équipe sera présent lors de ce premier TD de suivi
- 6 semaines de développement et suivi
- jeudi 29 avril 2010 : dernier TD de suivi
- dimanche 9 mai 2010 minuit : arrêt du développement (site redmine et svn)
- 11 et 12 mai 2010 : soutenance (planning à venir)
- ⚠️ L3 MIAGE / JNM, soutenance avancée au 10 mai 2010 matin
- ⚠️ L3 MIAGE / stage avant le 10 mai 2010, soutenance avancée à définir

❑ Evaluation

- 100 % en projet (pas de 2ème session)

Principe de suivi

❑ Jeudi après-midi :

- 13h30 – 15h30 : 2h en présence de l'enseignant tuteur
 - ◆ Questions sur les fonctionnalités à réaliser
 - ◆ Propositions / validation sur le découpage du travail
 - ◆ Validation sur la conception de l'application
 - ◆ Surveillance de l'avancement du projet
 - ◆ Suivi et évaluation de chaque membre de l'équipe individuellement
 - ◆ Aide technique sur le langage utilisé
 - ◆ Aide sur la réalisation (debug, mais pas trop...)

- 15h30 – 17h30 : salle réservée pour continuer à travailler en équipe
 - ◆ En profiter pour continuer d'avancer sur les points durs
 - ◆ Bien valider la répartition du travail
 - ◆ Finaliser un test d'intégration avec toute l'équipe

Soutenance et évaluation

☐ Soutenance (en mai)

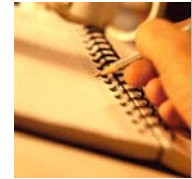
- Présentation (technique) de la réalisation
- Fonctionnalités réalisées (ou pas)
- Choix de conception (et justification)
- Mini-démo (attention, vraiment mini)
- Travail de chacun bien identifié
- Discussion sur les problèmes rencontrés et les solutions apportées
- Le tout en 15 minutes !

☐ Evaluation

- Sur la soutenance
- Le code, sa qualité
- L'utilisation des outils

Organisation

- ☐ Cours 1 : principes généraux - svn
- ☐ Cours 2 : Redmine et gestion de projet
- ☐ Cours 3 : Introduction à Eclipse
- ☐ Cours 4 : Eclipse C / PHP
- ☐ Cours 5 : V&V et tests unitaires, en Java
- ☐ Cours 6 : Tests C / PHP - conclusion

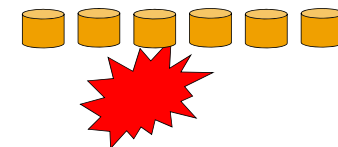


Gestion de version et de configuration

☐ Introduction à SVN

Motivations

- ☐ Quand on modifie des sources :
 - Des bugs apparaissent parfois (souvent !)
- ☐ On pourrait sauver chaque version de chaque fichier modifié...
 - Ou ne stocker que les différences !
- ☐ Et quand on est plusieurs à modifier
 - ☞ Savoir qui modifie quoi
 - ☞ Ne rien écraser
 - ☞ Fusionner si on modifie à plusieurs



Principe de la différenciation

- Outil diff
- Différences entre 2 fichiers d'après
 - Ligne de début/de fin
 - Insertion/Suppression d'une ou plusieurs lignes
- Facilité de détection et de construction d'un patch
- Pas de détection des lignes modifiées
 - Traitées comme suppression + insertion

Différenciation : illustration

```
WinMerge - [DiffContext.cpp - DiffContext.cpp]
File Edit View Merge Tools Plugins Window Help
Src\ - Src\ DiffContext.cpp - DiffContext.cpp
Location: g:\WinMerge\WinMerge_SVM\src\DiffContext.cpp | g:\WinMerge\WinMerge_268branch\src\DiffContext.cpp
CDiffContext::CDiffContext(LPCTSTR pszLeft /*=NULL)
: m_pFilterGlobal(NULL)
: m_pPluginInfos(NULL)
, m_nCompMethod(-1)
, m_bIgnoreSmallTimeDiff(FALSE)
, m_pCompareStats(NULL)
, m_pList(m_dirlist)
, m_pAbortable(NULL)
, m_bStopAfterFirstDiff(FALSE)
, m_pFilterList(NULL)
, m_pCompareOptions(NULL)
, m_pOptions(NULL)
{
  m_paths.SetLeft(pszLeft);
  m_paths.SetRight(pszRight);
  InitializeCriticalSection(&m_criticalSect);
}
/**
 * @brief Construct copy of existing CDiffContext
 */
CDiffContext::CDiffContext(LPCTSTR pszLeft /*=NULL)
: m_pFilterGlobal(NULL)
: m_pPluginInfos(NULL)
, m_nCompMethod(-1)
, m_bIgnoreSmallTimeDiff(FALSE)
, m_pCompareStats(NULL)
, m_pList(m_dirlist)
, m_pAbortable(NULL)
, m_bStopAfterFirstDiff(FALSE)
, m_pFilterList(NULL)
, m_pCompareOptions(NULL)
, m_pOptions(NULL)
{
  m_paths.SetLeft(pszLeft);
  m_paths.SetRight(pszRight);
}
/**
 * @brief Construct copy of existing CDiffContext
 */
```

Historique

- SCCS (livré avec Unix dès Vx, Bell labs programmer workbench, fusionné en 1983)
 - Delta descendant
- RCS (W. Tichy 1985)
 - Delta ascendant
- CVS (B. Berliner 1989)
 - WinCVS
 - Support dans beaucoup d'environnements...

Historique (suite)

- Subversion (subversion.tigris.org)
 - La relève de cvs
 - Bonne gestion des modifications de l'arborescence des répertoires
 - Installation et maintenance simplifiée
 - Nombreuses interfaces graphiques ou intégrations dans les IDE
- Visual Source Safe
 - The Microsoft Way
- ClearCase (Rational)
 - L'usine de gestion de traçabilité

Ce que SVN n'est pas...

- Un système de construction (makefile, ant...)
- Un système de gestion de projet (Ms-project)
- Un substitut à la communication entre développeurs (ex: conflit *sémantique*)
- Un système de contrôle du changement (bug-tracking, ChangeLog)
- Un système de tests automatisés
- Un système fondé sur un processus particulier

Commande(s) SVN

- `svn subcommand [switches] [cmd_args]`
 - Commande de base coté client
 - **subcommand** : obligatoire
 - **switches** : options spécifiques à la sous-commande
 - **cmd_args** : arguments de la sous-commande

```
svn checkout http://svn.c.net/rep/svn/trunk subv
```
- `svnadmin subcommand [switches] [cmd_args]`
 - Administration de la base

Base (ou dépôt) svn

- Locale (accéder directement par le client) :**
 - `file://`
- Accédée à travers Apache 2 (WebDAV)**
 - `http://`
 - `https://` (SSL encryption)
- Accédée par le protocole spécifique « svn » (possibilité de passer par ssh)**
 - `svn://` (nécessité d'avoir un serveur svnserv)
 - `svn+ssh://` identique à `svn://`, mais *tunneling ssh* (et pas de serveur)

Administrer une base SVN

- Créer une base SVN**
 - `svnadmin create /chemin/vers/referentiel`
 - **Par défaut format de stockage FSFS (autre format Berkeley-DB moins performant, conservé pour compatibilité)**



create

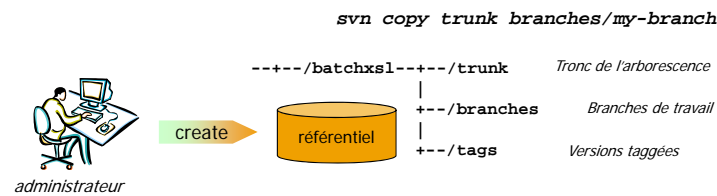


- `conf` : répertoire des fichiers de config
- `dav` : répertoire spécifique à `mod_dav_svn`
- `db` : les données (pas directement « lisibles »)
- `format` : un fichier avec un seul entier donnant le numéro de version des hooks de traitement
- `hooks` : répertoire des scripts de hook
- `locks` : répertoire des verrous de subversion
- `README.txt` : des infos sur les autres répertoires

Administrer une base SVN

- ❑ Au sein d'une base se trouvent un ou plusieurs projets.
- ❑ À chaque projet correspond en général un répertoire situé à la racine du dépôt et qui contient lui-même les fichiers et dossiers du projet.

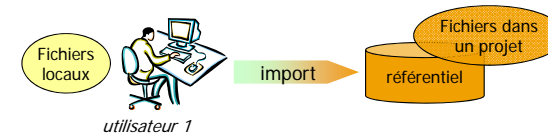
- Organiser les répertoires :



Importer des sources

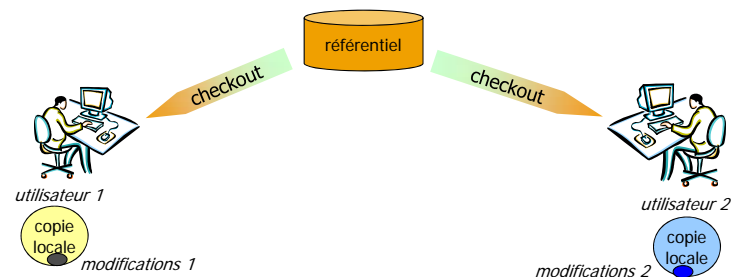
- ❑ Importer des sources

- `svn import rep_local /chemin/vers/referentiel [options]`
- `svn import myTree file:///usr/local/svn/newrepos/batchxsl/trunk -m "Initial import"`



Récupérer une copie locale des sources

- ❑ `svn checkout chemin/vers/referentiel/et/projet [options]`
- ❑ `svn checkout http://svn.collab.net/repos/svn/trunk`



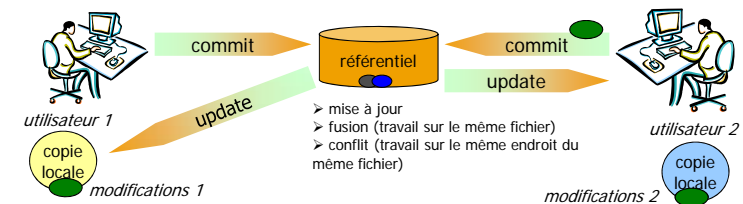
Propager ses changements Mettre à jour par rapport à la base

- ❑ Propagation de vos changements

- `svn commit`

- ❑ Récupération de nouvelles mises à jour

- `svn update`



Signification des sorties SVN pour *update* et *checkout*

- U file : **votre répertoire a été mis à jour**
- A file : **fichier ajouté à votre copie privée, sera propagé après commit**
- D file : **fichier effacé... définitivement après commit**
- C file : **conflit détecté lors de fusion**
- G file : **fusion effectuée (car pas de conflit)**

Quelques commandes et options

- Ajouter un fichier/répertoire** : `svn add`
 - + commit
- Retirer un fichier/répertoire** : `svn delete`
 - + commit
- Copie des fichiers/répertoires** : `svn copy`
 - + commit
- Déplacer des fichiers/répertoires** : `svn move`
 - + commit

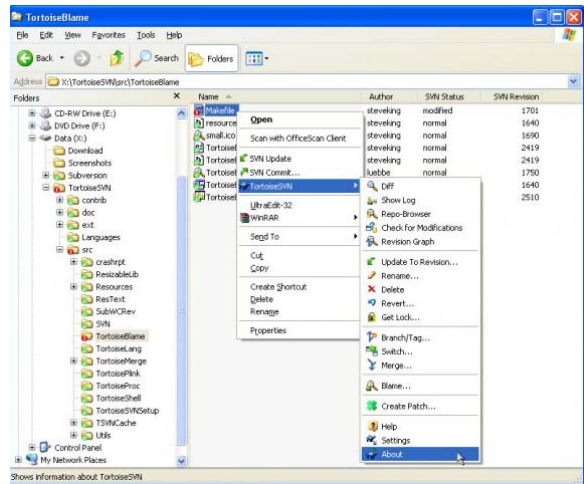
Quelques commandes et options (suite)

- Liste des répertoires dans le référentiel** :
 - `svn list`
- Affichage des messages de commit** :
 - `svn log`
- Mes modifications locales (pas de connection au référentiel)** :
 - `svn status`
- Visualiser les différences** :
 - `svn diff`
- Revenir en arrière (undo)** :
 - `svn revert`
- Indiquer qu'un conflit est résolu sur un fichier** :
 - `svn resolved sandwich.txt`

Plein d'autres choses...

- Les outils graphiques aident à rendre tout cela *vivable***
 - ☞ *RapidSVN (multi-plateformes)*
 - ☞ *TortoiseSVN (très bonne intégration dans Windows)*
 - ☞ *Les supports des environnements de développement*
 - ☞ *Eclipse : très bon support cvs, subclipse ou subversive pour svn*
 - ☞ ...

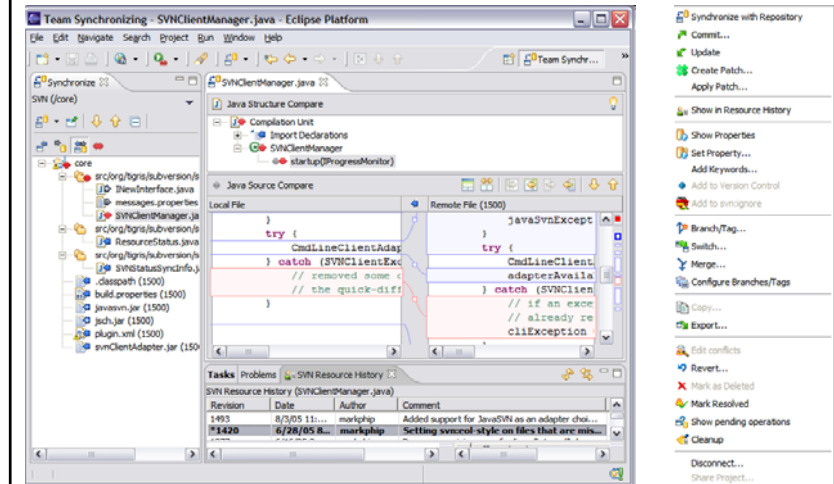
Illustration : TortoiseSVN



Ph. Collet

25

Illustration : plugin subclipse dans Eclipse



Ph. Collet

26

Références

- ❑ The SVN Book (en téléchargement, licence GPL)
 - <http://svnbook.red-bean.com/>
- ❑ Source d'information
 - <http://subversion.tigris.org/>

Ph. Collet

27

Mise en application

- ❑ Une base svn sera mise à disposition par projet
 - Connexion prévue grâce aux authentifications LDAP
 - Accessible depuis l'extérieur (j'espère !)
- ❑ Le tuteur a accès à la base (et au site de suivi Redmine)

Ph. Collet

28



Présentation des projets...