

Transmettre les évolutions d'un code source

1) Introduction

Un logiciel meurt lorsqu'il n'évolue plus. Cependant, canaliser le flux d'évolutions reste toujours le plus gros du travail. Celui-ci est souvent confié à des outils intégrés de gestions des évolutions comme CVS dans le cas de gros logiciels. Pour des logiciels plus modeste, la méthode manuelle est tout fait envisageable. Celle-ci passe par trois phases : la comparaison, la transmission et l'utilisation.

2) Comparer

Transmettre directement le code source modifié complet n'est souvent pas la bonne approche. Du côté du réceptionnaire, le code source a certainement évolué également, en récupérant le code complet, il perdra ses modifications. Nous allons donc fusionner les deux sources pour prendre ne compte les évolutions des deux côtés. Dans ce but, la prise en compte de modifications élémentaires est le plus simple. Celles-ci sont obtenues par comparaison. Deux sources ne peuvent être comparés que si les différences résultantes sont significatives. Une comparaison qui ressort des évolutions de formes comme un changement d'alignement ou de disposition n'est pas significative. Ainsi, le code source devra de préférence subir un toilettage avec un logiciel spécifique de mise en forme. Cela est vraiment appréciable lors d'un travail à plusieurs sur un texte, les différents styles de codage sont gommés par le reformateur. Seules les vraies différences apparaissent. D'autre part, la comparaison ne peut se faire que si la précaution de conserver le texte d'origine a été prise, soit nous pouvons dupliquer le répertoire contenant les codes sources dans son entier, soit nous pouvons dupliquer le texte source. Le répertoire ou le fichier dupliqué doit être au même niveau (lien symbolique possible), le nom est donc modifié en ajoutant une marque pour indiquer qu'il s'agit de l'original. Sans originalité, un zéro (0) ou "old" (ancien) est généralement ajouté.

La commande "diff" va produire les différences :

Entre deux répertoires et le contenu des sous-répertoires inclus :
Si certains fichiers sont seulement présents dans un des répertoires, des lignes avec "Only in ..." vont apparaître. Si certains fichiers sont des binaires, des lignes avec "Binary files ..." vont apparaître. Celles-ci sont supprimées.

```
$ diff -ur rep-old rep | grep -v "^Only in" | grep -v "^Binary files" > rep.diff
```

Entre deux fichiers du répertoire rep :

```
$ diff -u fich.old fich > fich.diff
```

que l'on peut automatiser par le script suivant pour tous les fichiers :

```
$ cat diff.sh
#!/bin/sh
for i in *.$1;
do if test -f $i.$2;
then diff -u $i.$2 $i;
fi
done
$ diff.sh adb old > fich.diff
```

On peut remplacer le paramètre de contexte unifié -u par -c plus ancien et affichant les différences dans un format moins lisible.

Quelques options supplémentaires permettent de "relâcher" la comparaison :

- i : ne prend pas en compte la casse du texte
- E : ne prend pas en compte les tabulations
- b : ne prend pas en compte les changements du nombre d'espaces
- w : ne prend pas en compte les espaces
- B : ne prend pas en compte les lignes vides
- N : considère un fichier absent comme vide

Celles-ci ne seront pas nécessaires avec l'utilisation d'un formateur automatique comme vu précédemment.

Il est bien sûr toujours possible d'éditer avec précaution (syntaxe à respecter) le fichier de différences pour éliminer les modifications à ne pas transmettre.

3) Transmettre

Le fichier contenant les différences doit être transmis en étant accompagné de la méthode d'utilisation de ce dernier.

Tout d'abord, nous devons indiquer dans quel répertoire se placer. Ce sera le premier niveau de répertoire où a eu lieu la comparaison.

Nous utiliserons la commande "patch" pour fusionner les modifications.

Nous devons alors indiquer les options de la commande "patch". Nous utiliserons l'option -N pour éviter de modifier un source contenant déjà la modification.

Nous utiliseront l'option -p1 dans le cas d'un fichier de différence issu d'une comparaison de répertoires.

Nous utiliseront l'option -p0 dans le cas d'un fichier de différence issu d'une comparaison de fichiers.

Cette dernière option supprime un nombre donné de niveau de répertoire inscrit dans le fichier de différence.

4) Utiliser

Une fois transmis le fichier contenant les différences, la commande "patch" va reconstituer le nouveau source à partir de l'ancien. Noter que cette commande utilise un heuristique assez élaboré pour retrouver l'emplacement de la modification. L'application de la modification peut ainsi s'appliquer sur un source qui n'est pas rigoureusement le source d'origine, mais plutôt à éviter.

Dans le cas d'un fichier de différence issu d'une comparaison de répertoires :

```
$ cd rep  
$ patch -Np1 < rep.txt
```

ou

Dans le cas d'un fichier de différence issu d'une comparaison de fichiers :

```
$ cd rep  
$ patch -Np0 < fich.txt
```

Patch indique alors la prise en compte des modifications avec quelques indications sur la méthode employée s'il rencontre des difficultés.

Pascal Pignard, octobre 2007.