Fabien DONIUS, Nicolas GRILL, Chérine KAMEL, Selim MILED - Ing1 Gr4

DOSSIER DE RÉALISATION DE CCE TESTEUR

Les codes correcteurs d'erreur

ECOLE CENTRALE D'ELECTRONIQUE
GROUPE ECE

I. Programme

Le logiciel devait être capable de lire un fichier, d'appliquer des erreurs, de les afficher, et enfin de les corriger.

D'autre part, il doit être possible de récupérer les informations pour les traiter par la suite.

Le principe de modularité nous ayant paru très important, la mise en place du système de plugins était obligatoire.

Nous nous sommes donc tournée vers un langage dans lequel, il est simple de créer une IHM mais tout en restant relativement bas niveau. Notre choix s'est donc tourné vers le C#.

a. Mise en place de la structure

Pour pouvoir mettre en place un système de plugin, il faut une interface entre les DLL plugin et le programme. Cette dernière fut la première chose de développer. C'est le centre névralgique de notre programme. Toute l'information passe par cette interface.

Nous avons donc déterminé la structure des DLL – voir ci-dessous – et avons pu nous pencher sur les fonctionnalités périphériques.

En effet, une fois ce système en place nous devons lui apporter les données, les afficher et enfin les enregistrer.

b. Ouverture de fichier

CCE testeur peut travailler sur deux types de fichiers : les textes et les images BMP. En effet un fichier texte ne comporte pas d'header et n'est pas compresser. Les BMP quant à eux possèdent un header de 54 octets et ne sont pas non plus compresser.

Pour pouvoir utiliser d'autre format d'images nous avons donc utilisé le Framework .NET pour faire une conversion de format. Ainsi facilement nous gérons une dizaine de format image.

c. Les données

Durant toutes les phases de traitement nous mesurons plusieurs informations. Celles-ci devaient être affichées, dans le programme. Les étapes de codage décodage et correction prenant un peu de temps nous avons du installer une progressbar qui informe l'utilisateur de l'avancement du traitement. D'autre part, Ce étapes étant gourmandes en ressource processeur, nous avons du séparer les phases de calcul et les phases de rapport d'information. Chacune sont dans ce que l'on appelle un thread, ce qui fait que notre application fonctionne parfaitement sur des processeurs double cœurs.

d. L'enregistrement

Toujours dans l'esprit de modularité, nous avons souhaité que l'utilisateur puisse personnaliser l'affichage des ses résultats en html. C'est pourquoi, il lui est possible de designer sa page.

D'autre part, vu le temps nécessaire au traitement de certains documents, il était impératif de pouvoir sauvegarder les résultats, pour ensuite les ouvrir de nouveau et les exporter dans un autre format.

Le format « *.ccerapport » est là pour ça. Il contient toutes les informations nécessaires pour pouvoir réafficher la simulation ultérieurement.

Les fichiers « *.ccerapport » sont composés comme suit :

- Sous forme de booléen : le type de fichier
- Sous forme d'entier : la taille du fichier source

- Sous forme binaire : le fichier source
- Sous forme d'entier : la taille du fichier erroné
- Sous forme binaire : le fichier erroné
- Sous forme d'entier : la taille du fichier corrigé
- Sous forme binaire : le fichier corrigé
- Sous forme de string : tous les autres paramètres et résultats.

e. L'aide

L'aide et les explications sont un point central pour comprendre les codes correcteurs. Si l'utilisateur ne l'a pas installée sur son poste, il sera redirigé vers notre site internet et la page correspondant à l'aide qu'il souhaite.

II. **DLL**

Les DLL sont composées de 3 classes. La première permet l'installation/désinstallation et la gestion des menus. La deuxième fait la même chose pour le bouton à coté de la sélection du plugin. La dernière sert au traitement des données.

Les DLL étant basées sur un système d'interface, elles peuvent être écrites dans tous les langages .NET.

III. **Installeur**

Pour réaliser l'installeur, nous avons utilisé NSIS. Cet outil nous permet de réaliser nos scripts d'installation personnalisés.

Pour CCE Testeur, trois points sont important : vérifier que le programme n'est pas déjà installé, vérifier que l'utilisateur dispose bien du .NET 2.0 et le cas échéant le télécharger et l'installer et enfin l'association des fichiers « *.ccerapport ».

Nous avons aussi séparé l'installation en trois parties : le logiciel (obligatoire ;-)), les fichiers d'aide et le devkit.

NSIS nous permet en plus l'écriture simple du logiciel de désinstallation.