

donc GROR pourrait être facilement amélioré en ce point.

 **Les contrôles:** il serait intéressant de permettre au concepteur d'une action système de placer des « warning » (un système de balise amélioré devrait permettre cela). Ceci afin d'afficher des avertissements lorsque des actions sont dangereuses (ex : rm -Rf). De plus, on peut imaginer un module de contrôle des actions recherchant certains « mots suspects »: les rm par exemple.

# Projet de gestionnaire de fichier sous LINUX.

## I. Idée de départ du projet:

Les environnements UNIX (dont LINUX) proposent de nombreux outils puissants en ligne de commande. Souvent, ces fonctionnalités ne se retrouvent pas dans les gestionnaires de fichiers classiques. De plus, l'utilisation de la ligne de commande est compliquée, peu conviviale et quelque peu fastidieuse. Le but de GROR est donc d'interfacer les outils de la ligne de commande afin de les rendre plus simples d'accès. GROR doit aussi permettre aux utilisateurs plus expérimentés d'automatiser des actions complexes.

## II. Ce qui a été fait:

- ☞ Un gestionnaire de fichiers dans lequel un utilisateur peu de manière graphique créer des lignes de commandes paramétrées qu'il pourra ensuite lancer sur un ensemble de fichiers (par simple sélection dans une liste et cliques de souris).
- ☞ La création d'un site (actuellement [www.gror.fr.st](http://www.gror.fr.st)) et d'un e-mail ([grorproject@hotmail.com](mailto:grorproject@hotmail.com)).
- ☞ Une documentation qui devrait permettre à l'équipe de reprendre le projet assez facilement. Le code source se veut le plus homogène possible et est commenté. De plus chacun des membres de l'équipe 2001/2002 se tient à la disposition de l'équipe 2002/2003 pour tout renseignements complémentaires.

## III. Ce qu'il reste à faire:

- ☞ **L'interface utilisateur** est très importante dans les logiciels actuels. D'autant plus pour une application telle que GROR dont le but est justement d'apporter cette interface manquante sous LINUX. Il semble que de grandes améliorations puissent être apportées sur ce point. NB : l'utilisation de QT est gratifiante pour le programmeur.
- ☞ Les principales méthodes concernant les opérations ensemblistes sur des listes de fichiers sont implémentées. Il reste donc à concevoir une interface graphique pour les exploiter.
- ☞ De même, il serait très intéressant de représenter graphiquement la création (chaînée) de sous-listes de fichiers au moyen de **filtres**: ces derniers pouvant être des actions créées par l'utilisateur.
- ☞ Le paramétrage de la ligne de commande peut être assoupli de manière à pouvoir effectuer encore plus de type d'action différentes (c.f. la classe ActionSyst et son interpréteur de ligne de commande).
- ☞ **Le fichier de configuration**: les actions sont lues à partir d'un fichier où elles sont codées sous forme de **balises**. Ce formalisme apporte des avantages indéniables au niveau de l'évolution (ex: apports de nouvelles balises) et des possibilités de commentaires. Ce principe n'a pas été entièrement exploité;