Programmation système
SmartFolder

David Gonzalez - Claudio Sousa

23 décembre 2016

1 Introduction

Ce TP de deuxième année en programmation système consiste à implémenter un programme similaire au SmartFolder sur MacOSX.

Le SmartFolder sur MacOSX recherche sur le disque des fichiers correspondant à un/des critères et pour chacun des fichiers trouvés, le programme crée un lien dans un dossier specifié.

1.1 Spécification fonctionnelle

Ce programme possède deux modes de fonctionnement.

1.1.1 Mode recherche

Le premier est le mode *recherche*. C'est le mode par défaut qui simule le SmartFolder sur MacOSX. Par ailleurs, le programme tourne indéfiniment tant qu'auncun signal d'arrêt n'est reçu.

Ce mode prend 3 paramètres :
- <*dir_name*> : chemin où stocker les liens ;
- <*search_path*> : chemin de recherche ;
- <*expression*> : critères de sélection.

<*dir_name*> et <*search_path*> sont de simples chemins vers des dossiers.
<*expression*> correspond à une liste de critères dont l'interface est un sous-ensemble à celle de *find*.

Expression
Comme dit précédemment, l'interface est semblable à celle de find. Voici la liste des options pris en charge :
- ... : ... ;

Output/comportement

Si des liens symboliques existent, ils doivent être suivis. Des fichiers a doublent ne doivent pas apparaitre dupliqués dans le dossier de destination.

1.1.2 Mode stop

Le deuxième est le mode *stop*. Il permet d'arrêter une recherche en cours.

Ce mode prend 1 paramètre :
- *-d* <*dir_name*> : termine le SmatFolder pour le chemin specifié.

2 Development

2.1 Architecture

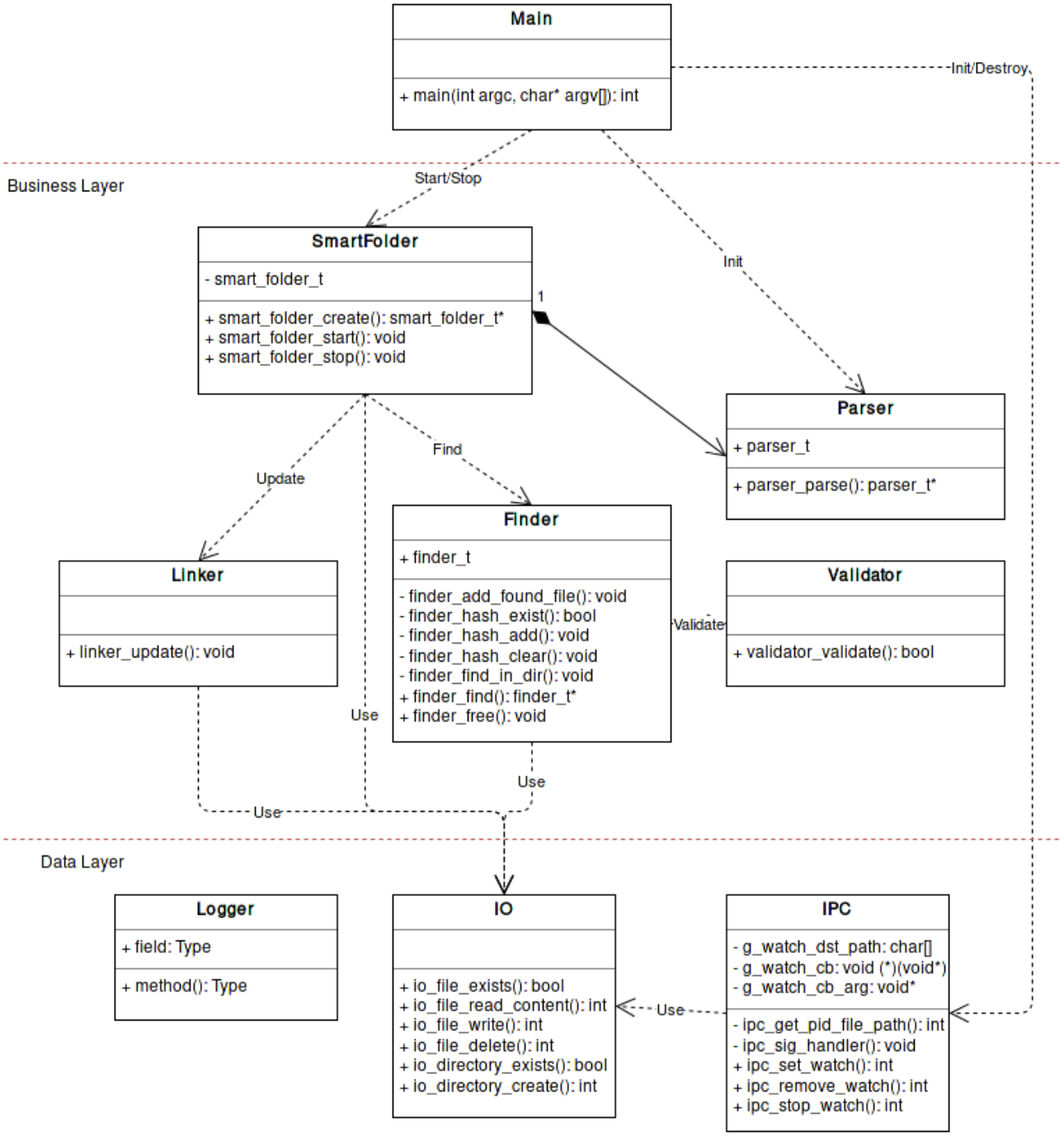

FIGURE 1 – Architecture du SmartFolder

2.1.1 Main

Le programme principal a pour rôle de vérifier les arguments et de sélectionner le bon mode de fonctionnement.
Dans le mode *recherche*, il a pour tâche de :

- met le processus en arrière-plan ;
- demande au module *Parser* de traiter l'expression ;
- initialise le module *IPC* ;

— initialise et lance le module *SmartFolder*.

Dans le mode *stop*, son seul rôle est de signaler l'arrêt à l'autre instance (voir IPC).

2.1.2 SmartFolder

SmartFolder est le module principal qui va orchestrer la recherche et la mise à jour du dossier de destination.

Lorsque lancé, il va continuellement utiliser le module *Finder* pour rechercher les fichiers correspondant au critère, puis donner la liste des fichiers retournée au module *Linker* pour qu'il mette à jour le répertoire de destination.
A noter qu'entre chaque recherche, il y a une pause de quelques secondes.

A l'arrêt, il est chargé de détruire le répertoire de destination et de libérer ses ressources.

2.1.3 Parser

Parser est le module qui transforme l'expression spécifiée dans la ligne de commande (critères de recherche) en une structure interne utilisable par le module *Validator*.

2.1.4 Validator

Ce module vérifie si un fichier est valide selon l'expression créée par le module *Parser*.

2.1.5 Finder

Ce module est responsable de produire la liste de tous les fichiers du dossier de recherche respectant les criteres de recherche.
La verification de l'expression est deleguée au module *Validator*.

2.1.6 Linker

Le module *Linker* a pour rôle de mettre à jour le dossier de destination à l'aide d'une liste de fichiers passée en paramètre.

Pour chaque fichier, il crée un lien si celui-ci n'existe pas. Il efface les liens qui ne sont plus valides.

2.1.7 IPC

Ce module est chargé de gérer la communication entre deux instances de *SmartFolder* et la logique derrière le mode *stop*.

La logique choisi est la suivante : au démarrage, l'application créer un fichier dans le répertoire utilisateur afin d'y stocker son PID et écoute le signal POSIX pour *SIGTERM*.
Lorsque une autre instance veut le stopper (mode *stop*), il lit ce fichier et lance le signal *SIGTERM* au PID lu.

Ce fichier se nommera d'après le chemin du répertoire de destination.
Lorsqu'il reçoit un signal d'arrêt, il libère la mémoire, efface le *pidfile* et declenche le stopage du smart_folder.

2.1.8 IO

Le but de ce module est d'offrir une interface simple aux appels systèmes etde factoriser la gestion des erreurs.

2.1.9 Logger

MOdule optionel qui factorise la gestion de logs si la fonctionalité est implémentée.