

Programmation concurrente

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 spécifié.

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'aucun 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 chemin vers des dossiers. Concernant *[expression]*, ceci correspond à une liste de critères dont l'interface est un sens-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 :

- ... : ... ;

1.1.2 Mode stop

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

Ce mode prend 2 paramètres :

- *-d* : paramètre qui permet de lancer ce mode ;
- *<dir_name>* : chemin traité par le SmartFolder à terminé.

2 Development

2.1 Architecture

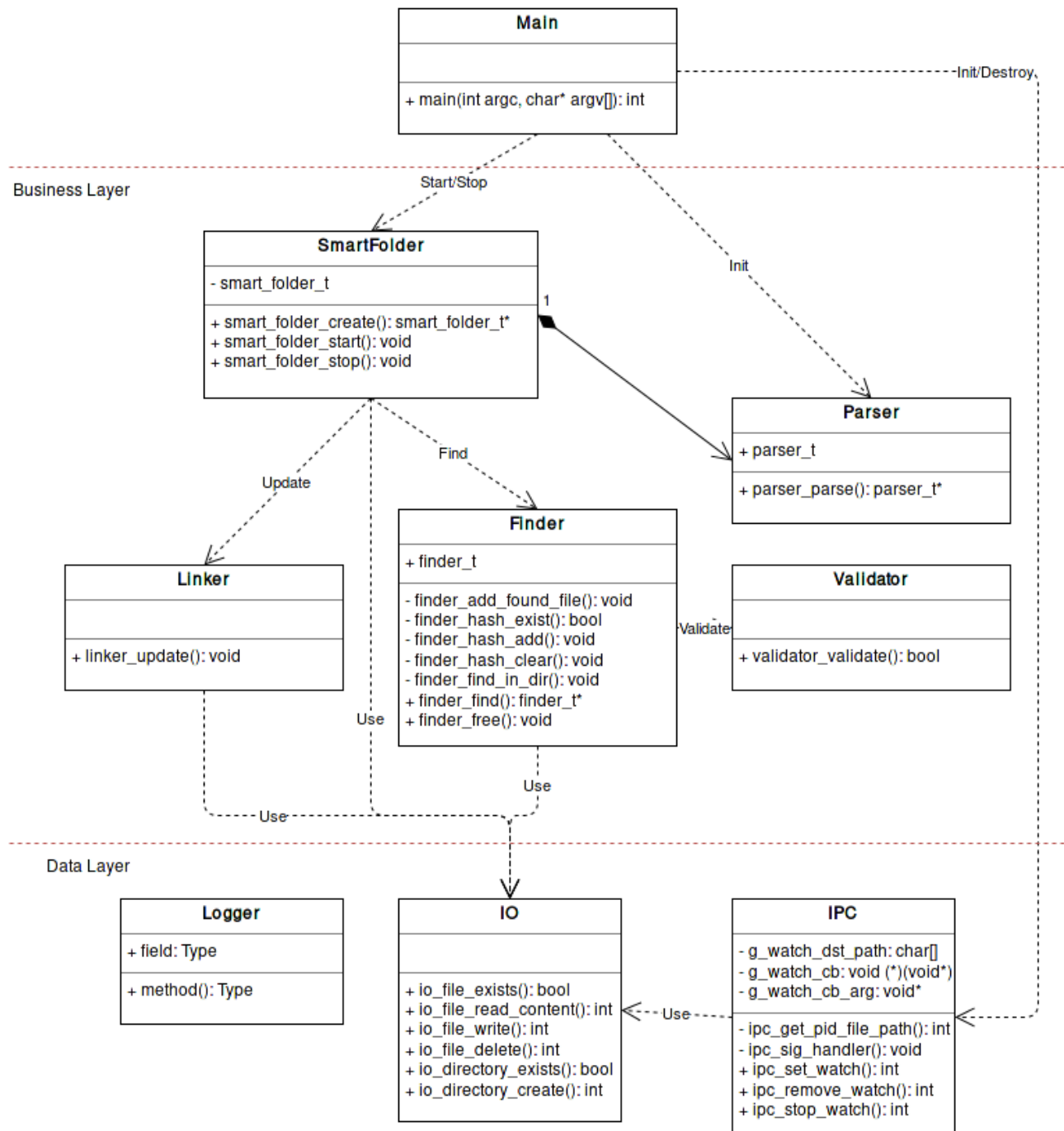


FIGURE 1 – Architecture du SmartFolder

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 :

- mettre le processus en arrière-plan ;
- traiter l'expression ;
- poser le *pidfile* ;

— initialiser et lancer la recherche.

Lorsqu'il reçoit un signal d'arrêt, il libère la mémoire, efface le *pidfile* et ce termine.

Dans le mode *stop*, son seul rôle est de signaler l'arrêt (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.

Lorsque un arrêt a été signalé, il est chargé de détruire le répertoire de destination et de libérer les ressources.

2.1.3 Parser

Parser est le module qui va s'occuper de prendre les paramètres de la ligne de commande correspondant à l'expression (critères de recherche), et de les transformer en une structure interne utilisable par le module *Validator*.

2.1.4 Validator

Ce module est responsable de la validation d'un fichier quelconque donnée en paramètre à l'aide de la structure interne créée par le module *Parser*.

2.1.5 Finder

La responsabilité de ce module est, en partant d'un dossier, de parcourir l'ensemble des fichiers/dossiers qui se trouve en dessous.

Pour chaque fichier trouvé, il doit le passer au module *Validator* afin que celui-ci détermine s'il correspond bien au critère de recherche.

Si le fichier est validé, alors il le stocke dans une liste à retourner.

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 fichier passé en paramètre.

Pour chaque fichier, il créera un lien si celui-ci n'existe pas.

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 choisie est la suivante : au démarrage, l'application crée un fichier dans le répertoire utilisateur afin d'y stocker son PID et enregistre un 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.

2.1.8 IO

Le but de ce module est d'offrir une interface simple aux appels systèmes.

2.1.9 Logger