

Active

– *Still active* –

Ce document a pour but de décrire les besoins, les exigences, les contraintes et les limites que nous allons devoir respecter durant la réalisation de notre projet.

Il résume les tâches que le programme doit pouvoir effectuer. Le document est composé de d'une première partie ouverte à tout lecteur et d'une partie technique qui explique les spécificités du produit plutôt destinée à un utilisateur averti.

Le document sert également de planning pour délimiter le temps nécessaire à la réalisation de chaque étape du projet et ce dans le but de le terminer à temps.

© Auteurs

NDJANDA MBIADA Jacques Charles¹
DIEP Anthony
DE OLIVEIRA Nelson
LEBEAU Christophe
JOOSSEN Matthieu

© Mentors

ZACCHIROLI Stefano
KESNER Delia

¹ Chef de projet.

SOMMAIRE

[Introduction](#)

[Qu'est ce qu'Active](#)

[Schéma d'utilisation](#)

[Interface graphique](#)

[Interface ligne de commande](#)

[Spécification](#)

[Structure de données](#)

[Architecture](#)

[Structurelle](#)

[Fonctionnelle](#)

[Base de données](#)

[Modules](#)

[Front End](#)

[UI](#)

[AQ Analyser](#)

[Active controller](#)

[C API \[Java, PHP, Python, OCaml, Ruby, Perl API ?\]](#)

[Back End](#)

[Performing model](#)

[Data base](#)

[File monitoring](#)

Introduction

Le progrès technologique est indéniable à notre époque, notamment en termes de capacité de stockage des ordinateurs personnels. L'un des enjeux des systèmes pour PC est la gestion des données utilisateur, à savoir permettre à ces derniers d'accéder rapidement à une information datant de plusieurs semaines, mois ou années sans y perdre des heures de recherche.

La capacité de nos disques durs peut atteindre aujourd'hui des centaines de gigaoctets (Go) et nous pouvons donc stocker des millions de fichiers sur un ordinateur personnel.

Un gestionnaire de système de fichier est un programme permettant d'accéder à ces données, d'y naviguer et de visualiser leur relation (quel fichier appartient à quel dossier par exemple). Ce programme n'a pas pour rôle de trier et de rechercher les fichiers et c'est pourquoi en l'état, l'utilisateur devra trier son disque dur à la main. Or, non seulement la machine sait mieux trier et rechercher que nous mais en plus elle le fait bien plus rapidement en cas d'oubli.

Bien que la plupart des systèmes d'exploitation répandus disposent en plus du gestionnaire de système de fichier un programme d'indexation, nous souhaitons écrire un programme alternatif au programme d'indexation natif de Linux / Unix et pourquoi ne pas le surpasser sur le plan technique et esthétique.

La réalisation du projet ne doit nécessiter aucun fond d'investissement, user uniquement de technologies open sources ou libres d'utilisation sans contraintes particulières sur le produit final. Pour palier à tout dépense supplémentaire (notamment sur le matériel de travail), nous avons à disposition des machines sous différentes plates-formes (NT, Linux, Unix) connectées à internet en très haut débit. Tout module permettant la réalisation du projet est inexistant en utilisation libre et devra alors être développé par l'équipe en charge du projet.

Le but est bien entendu de rendre le programme terminé, mais également de parvenir à produire ce dernier par une démarche professionnelle et de manière coopérative sous la direction d'un chef de projet. – La présentation du projet est faite ci-dessous. –

Qu'est ce qu'Active

Active est un logiciel de type surveillance et analyse de fichiers dans le but de faciliter la recherche de documents sur votre ordinateur.

Il existe actuellement des outils natifs de recherche intégrés à la plus part des systèmes d'exploitation : tel que Tracker, Spotlight ou encore Google Desktop. Active, à l'instar de ces programmes, vous permet d'accéder facilement au contenu stocké sur votre ordinateur. Il vous permet de retrouver les informations dont vous avez besoin et offre la possibilité d'y faire des recherches précises.

Utilisant une base de donnée légère et efficace, Active vous permet de rechercher un fichier quelconque sur votre ordinateur. Ceci grâce à un langage de requête simple et puissant. Son moteur de recherche flexible permet d'assurer l'interface entre la base de donnée et n'importe quelle interface accessible par l'utilisateur².

La pertinence et persistance des données de la base de données est assuré par un module en tâche de fond qui surveille les fichiers et scanne un grand ensemble du système de fichier aux moments propices.

Comme nous le verrons, l'efficacité d'Active repose sur sa modularité et sur le bon usage des méta-données des fichiers. La recherche sémantique que produit Active vous permet par exemple de trouver un fichier en effectuant une recherche par la date de dernière modification du fichier, son auteur système ou auteur de droit juridique en passant par la fréquence d'échantillonnage pour les fichiers audio par exemple.

² On notera que la pleine utilisation d'Active nécessite une première analyse complète du système de fichier

Schéma d'utilisation

Nous souhaitons ici détailler l'utilisation d'Active et distinguons pour ce faire deux grandes formes d'utilisations possibles.

Il est nécessaire de comprendre que l'utilisateur général peut aussi bien désigner un utilisateur d'outils informatiques non chevronné qu'un utilisateur expérimenté en système informatique.

Interface graphique

Le programme comporte une interface graphique qui est destinée aux utilisateurs qui ne maîtrisent pas la ligne de commande. Cette interface claire donne la possibilité d'effectuer une recherche sur le système de fichiers via l'outil d'indexation et de recherche de manière simple.

Cette interface épurée doit permettre la recherche de fichiers ou dossiers par nom, date de création, date de modification, propriétaire ou encore par taille de fichier. Elle doit permettre d'indiquer des intervalles de date de création ou de modification, de rechercher un fichier d'un certain type (ce type correspondant, à l'un des types de fichier énumérés dans la partie "spécifications techniques" du présent document : détaillant les structures de données du système), etc...

Ainsi que d'effectuer une recherche sur le contenu d'un fichier : l'interface doit permettre à l'utilisateur d'indiquer si sa recherche porte sur le contenu d'un fichier ou l'auteur juridique du fichier. Par exemple, une recherche sur une chaîne de caractères doit pouvoir retourner le compositeur de la musique d'un fichier audio ou son interprète.

Nous voulons en plus implémenter la recherche multiple, c'est à dire permettre à l'utilisateur de croiser des champs de recherche : rechercher un nom d'auteur de fichier PDF et de compléter cette recherche avec un intervalle qui correspondrait à la date de création du fichier par exemple. Cette fonctionnalité a pour but d'affiner la recherche et d'en améliorer la vitesse.

L'utilisateur général est l'utilisateur qui peut ne posséder aucune ou peu de connaissances en informatique. On doit, quoiqu'il arrive lui donner la possibilité de faire des recherches de manière simple sur son système de fichier via notre outil d'indexation et de recherche.

Il dispose d'une interface graphique lui permettant d'effectuer des recherches sur l'ensemble de son disque dur de manière simplifiée par rapport à la ligne de commande. Il doit également pouvoir utiliser la ligne de commande même s'il n'en maîtrise pas toutes les spécificités.

L'interface épurée dont dispose l'utilisateur novice lui permet la recherche de fichier, ou dossier

par nom du fichier, date de création, date de modification, propriétaire ou encore par taille de fichier. Le logiciel graphique doit par exemple lui permettre d'indiquer des intervalles de date de création ou de modification, d'indiquer si le fichier est d'un certain type (ce type correspondant à l'un des types de fichier énumérés dans la partie "spécifications techniques" du présent document ; détaillant les structures de données du système), etc..

De plus, il doit lui être possible d'effectuer une recherche sur le contenu du fichier : le logiciel doit permettre à l'utilisateur d'indiquer si sa recherche porte sur le contenu d'un fichier ou l'auteur juridique du fichier. Par exemple, une recherche sur une chaîne de caractères doit pouvoir retourner le compositeur de la musique d'un fichier audio ou son interprète.

Nous voulons en plus implémenter la recherche multiple, c'est à dire permettre à l'utilisateur de rechercher un nom d'auteur de fichier PDF et de compléter cette recherche avec un intervalle qui correspondrait à la date de création du fichier.

Cette fonctionnalité a pour but d'affiner la recherche et d'en améliorer la vitesse.

Interface ligne de commande

Le programme en ligne de commande doit offrir à l'utilisateur toutes les possibilités de l'interface graphique. Il offre à un utilisateur expérimenté la possibilité de ne pas avoir à lancer le programme graphique qui pourrait ralentir son travail s'il estime mieux maîtriser l'interface en ligne de commande.

L'utilisateur expérimenté peut donc utiliser la ligne de commande pour spécifier plus d'arguments que ce que l'interface graphique permet, ce qui peut contribuer à affiner grandement la recherche. Ou pour coupler le programme à un ensemble d'autre programme pour produire un nouveau programme.

Spécification

Cette partie sert à spécifier le fonctionnement du logiciel Active.

Elle comporte une description des interactions entre les modules ainsi que de l'architecture globale du projet, une description des structures de données et de leurs utilités et enfin une modélisation de la base de données et de sa spécification propre. Nous adoptons un point de vue interne pour ces descriptions.

Structure de données

On utilise :

- une énumération FileTypeEnum pour les différents types de fichier :
 - o - FileTextEnum : comprend les extensions de fichiers textes
 - o - FileSoundEnum : idem pour les fichiers audios
 - o - FileVideoEnum : idem pour les fichiers vidéos
 - o - FileObjectEnum : extension pouvant avoir des images, sons etc. e.g doc ppt ...
 - o - FileDirEnum : pour distinguer les répertoires (à voir)
 - o - FileOtherEnum : Il existe des fichiers qu'on sait être autre e.g .so, .a a.out ...
-

[A suivre ...]

Architecture

Active est constitué de plusieurs modules dont les actions coordonnées permettent le fonctionnement du programme. C'est ainsi qu'il est possible de retrouver des fichiers sur un ordinateur compatible et équipé d'Active. Son architecture structurelle et fonctionnelle doit permettre un certain niveau de modularité. L'utilité et le fonctionnement de ces modules sont détaillés dans la partie "Module" de ce présent document.

Nous nous concentrons ici sur l'architecture interne du programme général en nous appuyant sur les structures décrites dans la partie précédente et précisons les modules auxquels sont liés les spécifications présentées.

Structurelle

La structure du programme est un point primordial.

Nous demandons une certaine qualité en terme de modularité du programme final. C'est pourquoi Active est structuré autour de plusieurs parties.

- Une base de données dans laquelle tous les fichiers présents sur le disque dur sont répertoriés et régulièrement mis à jour ;
- Un programme qui est actif en tâche de fond et se charge d'alimenter la base données ainsi que de la tenir à jour en cas de modification d'un fichier déjà enregistré ;
- Un programme de recherche de fichiers dans la base de données ;
- Un autre programme capable d'informer le précédent programme d'interroger la base de données (IV) ;
- Un programme (V) interprétant les requêtes d'un usager pour transmettre au programme IV, qui de suite va retransmettre l'information lui aussi.
- On se réserve la possibilité selon le planning fixé de créer un programme graphique capable de générer les requêtes attendu par le programme (V).

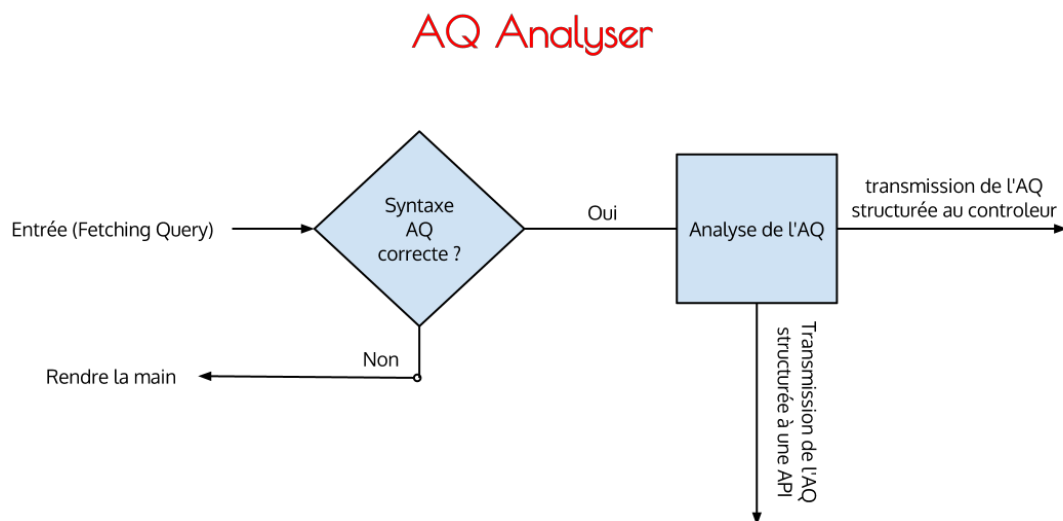
Fonctionnelle

[La fonction de recherche prend en arguments des mots-clés qui sont employés pour retrouver des fichiers dans la base de données.]

[[Les modules sont à détaillés ici j'ai mis un petit texte pour comprendre les roles](#) donc le modèle vous n'avez pas compris son role (c'est le design patern MVP).

Bref, ici il faut expliquer comment interagissent les modules avec des mots et les diagrammes à faire.]

L'analyser



© Active

Le contrôleur :

- vérifie si la requête utilisateur correspond à la syntaxe AQ
- si oui, la requête utilisateur doit être insérée dans une structure struc_req_util contenant les informations organisées (si non, sortie du programme)
- cette structure est alors lue pour la transformer en requête de base de données
- la base de données retourne le résultat qui est alors inséré dans une liste de struc_res_bdd
- cette structure est alors transformée dans un langage (type JSON, html)
- la lecture de ce fichier permet d'afficher le résultat de la requête

Le modèle :

- reçoit les requêtes du contrôleur
- interroge la base de données
- retourne le résultat au contrôleur

La base de donnée

Le démon

Potentiel interface graphique

Base de données

Nous avons opté pour l'utilisation de SQLite comme gestionnaire de base de données. Ce choix est justifié par la légèreté et donc dextérité de ce SGBDR.

Modules

[Explication détaillée point de vue développeur remarquez qu'en une phrase ces modules sont expliqués]

Front End

Les modules au devant de la scène. Modules facilement accessibles par d'éventuelles API.

UI

J'ai bien dit UI pas GUI et pour le moment l'UI est le front-end avec AQ et les arguments comme possibilité de commande

AQ Analyser

Le front-end entre l'homme et le programme

[Explication détaillée point de vue développeur]

Active controller

Reçoit une requête, fait ses analyses, transmet au modèle, reçoit les données et informe la vue

C API [Java, PHP, Python, OCaml, Ruby, Perl API ?]

Ensembles de classes et fonctions pour manipuler le programme Active.

Back End

Module en arrière plan cachés au mieux invisible au possible.

Performing model

Permet la communication entre les autres modules d'Active et la base de données.

Data base

Gestion de la base de données, données persistantes, etc.

File monitoring

Daemon qui scanne tout le système de fichier régulièrement et intelligemment ; met à jour la base de donnée grâce au modèle. Voilà le seul programme lancé au démarrage du système.

