

PROJET FIL ROUGE - 2019-2020

I- Qu'est-ce que le projet « Fil Rouge » ? *** RAPPEL ***

Il s'agit d'un projet transversal ayant un double objectif :

- valider des **compétences scientifiques et techniques** liées aux concepts, méthodes, et langages abordés en 1^{ère} année de la spécialité Systèmes Robotiques et Interactifs (SRI).
- valider également des **compétences générales** en lien :
 - avec la **gestion de projet** aussi bien sur le plan technique (différentes phases de mise en œuvre d'un projet) qu'en terme d'**organisation** (comment fonctionner en groupe et avec le client),
 - avec la **communication** (savoir restituer/présenter son travail).

I-1. Des compétences scientifiques et techniques ...

Le détail des compétences techniques mises en jeu dans ce projet, ainsi que les noms des intervenants sont donnés dans le tableau 1.

UE et Semestres Concernés	Matières – Compétences Objectifs généraux	Enseignants & Heures associées
S5 - Informatique	Programmation en C, commandes et script UNIX (réutilisation de fonctions développées en TP – Gestion d'une pile - approfondissement C et UNIX – manipulation des fichiers, lecture-écriture formatées, manipulation de chaînes de caractères, ...) - Traitement de base de documents de type texte, image, audio	Isabelle Ferrané & Julien Pinquier <i>Heures projet + travail personnel</i> Julien Vanderstraeten <i>TD dédiés</i>
S5 - Outils de modélisation informatique	Modélisation UML Mise en pratique UML – diagrammes de classes et de séquences	Christelle Chaudet <i>Application du cours</i>
S5 - Ingénierie des systèmes	Sûreté de fonctionnement - Développement de logiciels sécuritaires	Julien Vanderstraeten <i>Application du cours</i>
S6 - Gestion de Projet	Spécification, conception, développement, tests et intégration	Julien Vanderstraeten <i>Heures projet + travail personnel</i>
S6 - Conception orientée objets	Conception et Modélisation Objet - UML Java/Eclipse Application directe des concepts objets Design pattern Modèle-Vue-Contrôleur, Observable-Observer, ...	Christelle Chaudet Julien Vanderstraeten <i>Application du cours + travail personnel</i>
S6 - Introduction aux Systèmes Robotiques et Interactifs – partie IHM	Prototypage rapide, processing, ... Application directe des concepts IHM	Philippe Truillet <i>Application du cours + Travail personnel</i>
S6 - Introduction aux Systèmes Robotiques et Interactifs – partie Robotique, Image, Son	Arduino, capteurs, traitement image, traitement audio, ...	Michel Taïx & Michaël Laeur <i>Qqs TP dédiés + heures projet + travail personnel</i>

Tableau 1 : Compétences techniques

I-2. ... aux compétences plus générales ?

Ce projet est l'occasion de décroisonner les enseignements et de faire le lien avec d'autres domaines de compétences (voir tableau 2). Dans ce contexte, l'organisation de l'équipe, la gestion du projet, la communication avec le client, la qualité des présentations écrite/orale seront prises en compte dans la notation, selon une grille qui vous sera communiquée par les intervenants et mise à disposition sous Moodle.

UE et Semestres Concernés	Matières – Compétences Objectifs généraux	Enseignants & Heures associées
S5 - Sciences humaines et sociales	S'organiser pour réaliser un projet en équipe. Savoir convaincre un jury, des clients (communication orale/écrite). Réaliser un retour d'expérience et capitaliser ses compétences (techniques et non techniques).	Geneviève Estadiou estadiou.g@gmail.com Cours et TD de communication (S5) + coaching(S6)

Tableau 2 : Compétences générales

I-3. Organisation générale et évaluation

Ce projet est organisé en 3 parties, chacune ayant pour objectif de mettre en pratique, d'approfondir et de valider des compétences bien précises (voir Tableau 3).

Décomposition du projet	Focus	Points faisant partie de l'évaluation
Partie I : Indexation de documents et moteur de recherche	Spécifications Développement en langage C, commandes et scripts Unix Programmation modulaire Traitement de documents de type texte, image, son Intégration, test et validation	<ul style="list-style-type: none"> Document de spécifications Validation sur machine Implication individuelle Rapport/Bilan sur la partie I (partie technique & partie communication)
Partie II : Moteur de recherche et Interface utilisateur avancés	Conception OO et application de design pattern Interfaçage langage C et Java Prototypage de l'interface Intégration, test et validation	<ul style="list-style-type: none"> Dossier de conception Implication individuelle Présentation sur la partie II (partie technique & partie communication) Démonstration
Partie III : Application à la robotique	Développement de fonctionnalités en lien avec la coupe de France Robotique	<ul style="list-style-type: none"> Démonstration Présentation Qualité du code

Tableau 3 : Organisation du projet

Ce document présente par la suite le cahier des charges de la deuxième partie du projet Fil Rouge ainsi que le calendrier associé.

II- Moteur de recherche et Interface utilisateur avancés

Il s'agit de faire évoluer le moteur réalisé dans la première partie du projet et de le doter de fonctionnalités avancées.

- (1) Une interface intégrant des **aspects graphiques** ainsi que différents **modes d'interaction**. L'objectif est de faciliter et de rendre robuste l'interaction de l'utilisateur avec le système. Cela concerne aussi bien la saisie des requêtes l'affichage des réponses et la consultation des documents sélectionnés.
- (2) La possibilité, non seulement de prendre en compte des **utilisateurs standards**, mais également un **utilisateur privilégié** (administrateur) autorisé à effectuer des tâches de configuration de l'application.
- (3) de la possibilité de faire des **requêtes complexes** et d'effectuer une même requête sur un ou **plusieurs moteurs de recherche**. Les résultats des différents moteurs de recherche peuvent alors être comparés, fusionnés, ...

III- Lien entre la partie I et II du projet

La **partie 1** du fil rouge a pour objectif de développer en langage C et scripts Unix, les fonctionnalités d'indexation et de comparaison de descripteurs ainsi qu'une version basique du moteur de recherche et de l'interface. **Les fonctionnalités d'indexation et de comparaison ne doivent pas être recodées dans la partie 2, mais utilisées telles quelles.** Pour cela il faudra interfacer les langages Java et C. Plusieurs solutions sont envisageables pour réaliser cet interfaçage. Une des solutions de cet interfaçage se traduit par la mise en œuvre de la JNI (Java Native Interface). Un TD dédié est programmé en mars. Vous devrez justifier vos choix technologiques.

Les fichiers résultant de la phase d'indexation (*base_descripteur*, *liste_base*, ...) sont exploités par le moteur de recherche. L'interface minimaliste en mode console développée dans la partie 1 sera remplacée par l'interface graphique implémentée dans la partie 2 afin d'élargir les possibilités d'interaction et de recherche.

IV- Evaluation et Validation de la Partie II

L'évaluation tiendra compte de plusieurs critères. Un **dossier de conception** appliquant les principes de la modélisation UML devra être rendu avant de démarrer le développement. Ce document devra présenter les **cas d'utilisation** que vous aurez identifiés dans votre analyse du problème ainsi que les **diagrammes de séquence, d'états et de classes**.

A l'issue du projet, **une présentation**, suivie d'une **démonstration** et d'une **séance de questions** permettront d'évaluer les aspects communication (aspect « commercial », c'est-à-dire mise en valeur des compétences acquises, savoir communiquer sur un

produit), techniques (choix de conception, maîtrise du sujet, difficultés rencontrées) et fonctionnels (démonstration, validation).

V- Planning Partie II

Le calendrier ci-après vous indique les différentes échéances de cette partie du projet.

Organisation	2019-2020
Débriefing de la validation de la Partie I (Gestion de projets)	27/01/2020 J. Vanderstraeten - U3-salle 2
Mise à disposition du cahier des charges de la Partie II	Semaine du 10/02/2020
Lancement de la Partie II (Gestion de projets)	10/02/2020 J. Vanderstraeten
TD Fil Rouge	14/02/2020 J. Vanderstraeten
Début des TP COO	13/02/2020
Dépôt du dossier de conception + Planning prévisionnel + répartition du travail au sein de chaque groupe.	08/03/2020 sur Moodle
Dépôt d'une première version moteur avancé avec interface textuelle	22/03/2020 sur Moodle
Préparation de la Validation (Gestion de projets)	01/04/2020 J. Vanderstraeten
Dépôt d'une version beta de l'application	05/04/2020 sur Moodle
Validation technique + Communication + Gestion de Projets	21/04/2020 après-midi – A définir I. Ferrané, J. Piquier, J. Vanderstraeten, P. Truillet, G. Estadiou & A. de Bonneval C. Chaudet
Dépôt de la présentation	22/04/2020 sur Moodle
Dépôt version finale et dossier	02/05/2020 sur Moodle

Les consignes détaillées relatives aux évaluations seront précisées ultérieurement. Ce planning est susceptible d'être modifié, notamment les dates des séances de TD Gestion de projet dédiées.

ANNEXE : Cahier des charges – PARTIE II

Consignes : Le moteur de recherche principal sera celui qui a été développé dans la partie I. **Les groupes resteront donc les mêmes.** Vous serez aussi susceptibles d'utiliser d'autres moteurs réalisés par d'autres groupes.

1- Objectifs

L'**objectif pédagogique général** de cette partie du projet consiste :

- (1) à **réutiliser** des concepts vus en cours d'IHM (avec P. Truillet) et de conception orientée objet et UML (avec C. Chaudet)
- (2) et à les **appliquer** au **contexte** du projet de moteur de recherche sur lequel vous avez déjà réfléchi et pour lequel vous avez développé des solutions.

L'**objectif technique** consiste à rendre le **moteur de recherche** facile d'utilisation, et à le doter de fonctionnalités avancées. L'approche MVC (Modèle-Vue-Contrôleur, cf. Schéma 1) présentée dans le cours de Conception Orientée Objet et incluant le design pattern Observer-Observable permettra de développer une application dotée d'une **interface graphique** permettant aux utilisateurs d'interagir avec un ou plusieurs moteurs. D'autres design patterns pourront également être utilisés.

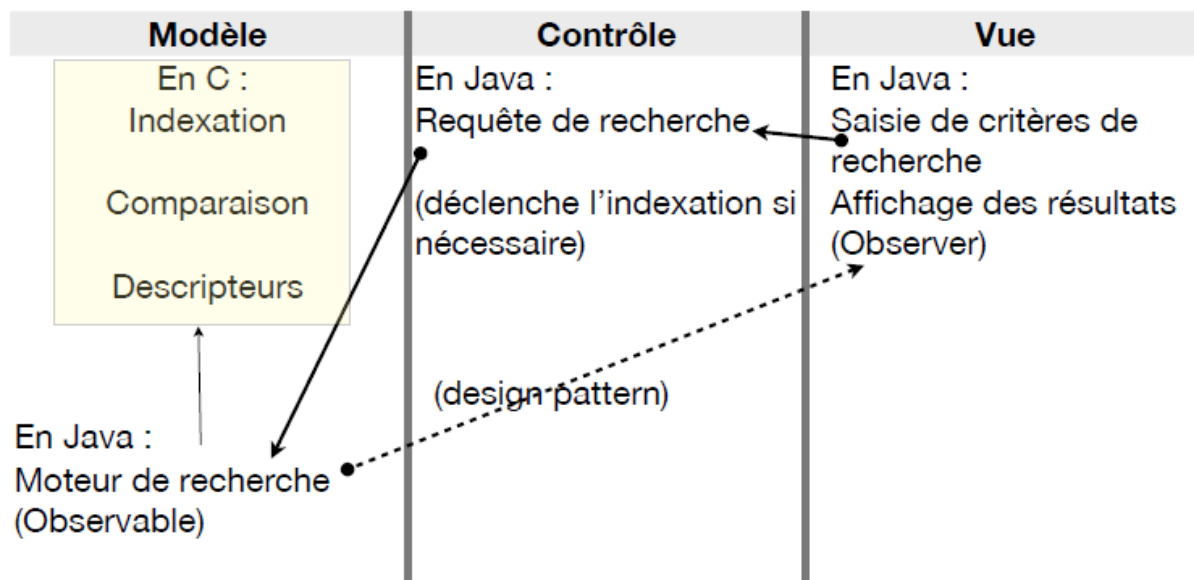


Schéma 1 : Modèle MVC présenté dans le cours de COO et appliqué en TP de COO

2- Fonctionnalités avancées

L'interface doit permettre à l'utilisateur de définir les éléments qui constitueront la requête qui sera envoyée au moteur. Une requête est constituée d'un ou de plusieurs **critères** permettant de rechercher des documents dans les bases de documents préalablement indexées (cf partie 1 de projet).

2.1- Lancement d'une requête

Dès le lancement de l'application, il est proposé à l'utilisateur de procéder à la recherche de documents en lui permettant de spécifier un ou plusieurs critères de recherche. Plusieurs types de requêtes pourront alors être réalisés. Les recherches pourront être faites via un ou plusieurs moteurs de recherche (voir détail section 2.2).

2.1.1- Critères de recherche

Compte tenu de la nature des documents, un critère de recherche peut correspondre soit à :

- un mot clé.

Dans ce cas, tout texte dont le descripteur contient ce mot-clé est une réponse potentielle. Le nombre d'occurrences du terme peut aussi faire partie de la requête. Lors d'une recherche textuelle la validité d'un mot-clé sera vérifiée avant de lancer la recherche. Un test de validité pourra par exemple rejeter les suites de caractères non pertinentes comme « &** !! »,

Exemple : rechercher les documents dans lesquels le mot-clé « Musique » apparaît plus de 3 fois.

- une plage de couleurs dominantes.

Dans ce cas, toutes les images dont les descripteurs ont une valeur supérieure à un seuil dans cette plage de couleur correspondent à une réponse potentielle.

Exemple : rechercher les images dans lesquelles la plage de couleur « Bleu » domine à 40%.

- un enregistrement audio (jingle, événement sonore, ...).

Dans ce cas tous les enregistrements audio du corpus qui contiennent cet extrait sonore plus de X fois constituent une réponse potentielle. Le nombre de fois que ce son apparaît peut aussi faire partie de la recherche.

Exemple : rechercher les documents sonores qui contiennent au moins 2 fois cet extrait.

- un fichier utilisé comme exemple.

Celui-ci peut appartenir à la base de documents ou bien être extérieur à celle-ci (dans le respect des formats pris en compte). Dans ce cas l'utilisateur fournit/sélectionne le nom du fichier à utiliser comme exemple. Le moteur recherche alors les documents les plus proches (dont la similarité est supérieure à un seuil). La recherche de documents proches ne sera lancée que si le fichier requête existe bien et est au bon format. Dans le cas contraire l'utilisateur sera informé du type d'erreur.

Remarque : on peut avoir des curseurs pour régler les seuils ... des listes pour sélectionner un nom de fichiers ... Une palette pour la sélection de couleur ... Tout ce qui peut améliorer la robustesse (limiter les erreurs de saisie et les incohérences).

2.1.2- Requête simple

Un utilisateur peut formuler une requête simple constituée d'un critère de recherche (comme dans le moteur basique de la première partie) et d'un seuil de sélection.

2.1.3- Requête complexe

Une requête complexe est constituée de plusieurs critères de recherche. Les critères de recherche peuvent avoir une polarité (+) ou (-). Dans un premier cas, le critère doit être présent dans le résultat et dans le second cas il doit en être absent. La polarité par défaut est (+).

Exemple de requête portant sur plusieurs mots-clés :

Football Mondial Argentine

Equivalut à (+Football +Mondial +Argentine)

→ on recherche des textes dans lesquels les 3 mots clés sont présents.

Football -Mondial

→ on recherche des textes avec le mot clé football mais sans le mot clé Mondial

Il peut en être de même pour les autres critères. On peut chercher des couleurs ou des sons qui n'y sont pas ou des fichiers qui sont complètement différents du fichier fourni.

2.2- Phase de recherche

Une fois la **requête validée** (cas d'erreurs identifiés et écartés) la recherche est lancée. L'interface indique à l'utilisateur que le système procède à la recherche (sablier, barre de progression ou équivalent...) et le **moteur développé dans la partie 1** du projet doit être utilisé.

ATTENTION : Si votre moteur n'est pas complètement fonctionnel, vous simulerez le résultat de la recherche de façon à illustrer quand même les différentes situations. Vous construirez alors un jeu de données approprié et significatif. Les modifications et corrections du moteur développé dans la partie 1 ne seront pas prises en considération dans la notation de la partie 2.

2.2.1. Recherche multi-moteurs

L'utilisateur peut sélectionner le moteur de recherche dans une liste de moteurs disponibles. Une requête lancée sur un moteur peut être lancée sur un autre. Les résultats peuvent être comparés ou fusionnés (efficacité, fiabilité...).

Une requête peut être lancée en parallèle sur plusieurs moteurs de recherche. Les résultats des différents moteurs de recherche sont alors consolidés pour fournir à l'utilisateur les résultats les plus pertinents. Les règles de consolidation sont basées sur des systèmes de vote.

2.2.2. Recherche en mode ouvert ou fermé

La **recherche en mode fermé** suppose que l'on accède seulement aux descripteurs existants. Il n'y a pas de mise à jour possible au fil de l'eau.

La **recherche en mode ouvert** suppose que si en cours d'utilisation du moteur de nouveaux fichiers étaient ajoutés, ils seraient automatiquement indexés et pourraient apparaître dans les résultats de recherche des requêtes suivantes.

2.3. Différents acteurs

Si l'indexation de nouveaux documents doit être effectuée, cela doit être complètement transparent pour l'utilisateur. Les paramètres d'indexation comme les seuils, ne sont connus et modifiables que par l'**administrateur**. L'**utilisateur standard** n'en a pas connaissance. Il convient donc de séparer (onglets différents, ...) les fonctionnalités de recherche proposées à l'utilisateur standard, des fonctionnalités de « réglage » que peut opérer l'administrateur sur le processus d'indexation.

2.4. Résultat d'une requête

Les résultats d'une requête doivent pouvoir être affichés suivant différentes modalités et sauvegardés à la demande de l'utilisateur.

2.4.1. Modalité d'affichage et consultation des résultats

L'interface affichera le **résultat correspondant à la recherche** effectuée, les documents étant classés par **ordre de pertinence** (du plus proche ou moins proche).

L'affichage doit pouvoir se faire sous différents modes : liste de liens cliquables, d'icônes, d'images, ... correspondant aux documents trouvés dans la base de documents indexés. Par exemple pour le mot clé **musique**, vous aurez une **liste ordonnée de fichiers** correspondant à ce mot-clé, liste que vous proposerez comme résultat de recherche, idem pour les autres types de documents.

L'utilisateur pourra **sélectionner un document** en cliquant sur un item de la liste proposée. Le système permettra suivant le cas, la visualisation (texte/image) ou

l'écoute (audio) du document sélectionné. Si trop de fichiers répondent à la requête, une stratégie d'affichage progressive pourra être définie (affichage par bloc de X fichiers par exemple).

2.4.2. Gestion des erreurs

L'utilisateur doit être informé, de manière appropriée, des différentes situations d'erreurs ou de mauvaise utilisation du système. Exemple : *absence de résultat, critère de recherche non valide, format de fichier exemple non compatible, ...*

2.4.2. Mémorisation des résultats

Durant une session, l'utilisateur peut demander à sauvegarder les requêtes et résultats associés. L'utilisateur peut alors y revenir sans que la requête ne soit relancée.

3- Dossier de conception (ou dossier mi-parcours)

Le problème doit être analysé et modélisé. Les enseignements de COO doivent vous permettre de pouvoir passer de la modélisation UML au code. Le dossier de conception est à rendre pour le **08/03/2020**. Celui-ci doit être accompagné :

- d'un planning prévisionnel concernant la phase de développement, de test et d'intégration ;
- ET d'une répartition équilibrée des tâches. **Chaque membre du groupe doit être responsable d'une tâche bien identifiée.**