



Réponse à l'appel d'offre du :

META-MOTEUR DE RECHERCHE SUR LE WEB ORIENTE COMMUNAUTE

Équipe :

M. Jérémie FRÉCHARD

Mlle Cécile GIRARD

Mlle Aysel GUNES

M. Pierre RAMOS

Étudiants en Master Informatique première année

Client :

M. Rushed KANAWATI

Encadrant :

Mme Sophie TOULOUSE

(17/02/2006)

Version 1.0

Sommaire

I/ Analyse du contexte.....	3
1) Analyse du besoin.....	3
a) Introduction.....	3
b) Objectif du projet.....	3
c) Type d'utilisateur.....	3
d) Contexte d'utilisation.....	4
2) Analyse de l'existant.....	4
 II/ Réponse aux besoins.....	 5
1) Fonctionnalités.....	5
2) Préconisations techniques.....	6
a) Architecture globale et intégration au SI client.....	6
b) Techniques mise en œuvre.....	6
c) Exigences opérationnelles.....	6
3) Investissement matériel et logiciel.....	7
4) Impacts organisationnels.....	7
5) Moyens fournisseurs.....	7
a) Moyens matériels et logiciels.....	7
b) Moyens humains.....	7
 III/ Organisation.....	 8

I/ Analyse du contexte

1) Analyse du besoin

a) Introduction

Le projet porte sur le développement d'un Méta-Moteur de recherche sur le Web orienté communauté. Les moteurs de recherche sur le Web, comme Google, AltaVista ou autres, sont aujourd'hui des outils incontournables pour aider les utilisateurs à chercher et localiser l'information sur le Web.

La popularité de ces outils est en grande partie due à leur simplicité d'utilisation. En effet, les utilisateurs soumettent des requêtes, souvent formés de quelques mots clés et le moteur retourne les documents pertinents pour la requête formulée. Les performances actuelles des moteurs existants en termes de précision et de rappel sont encore assez médiocres. A titre d'exemple la précision de Google ne dépasse les 29% et le rappel est de 20% seulement en 2004. Si le faible rappel est souvent justifié, à juste titre, par la grande taille du Web et sa rapidité de croissance, la faible précision est due en premier lieu à la faible spécification des requêtes utilisateurs.

Différentes approches sont proposées dans la littérature scientifique afin de répondre à ce problème. Dans ce projet, nous nous intéressons à une approche dite « communautaire ».

b) Objectif du projet

L'objectif de ce projet est de permettre à un groupe d'utilisateurs, qui forment une communauté d'intérêts, de partager leurs expériences de recherche d'information d'une manière implicite et intelligente. En effet, dans une communauté d'utilisateurs (par exemple les membres d'un laboratoire de recherche), l'intersection entre les thèmes de recherche sur le Web peut être assez importante. Par conséquent, un moteur de recherche qui garde la trace des requêtes passées soumises par des utilisateurs peut en profiter afin d'améliorer ses performances en filtrant les réponses obtenues de sorte à promouvoir les ressources approuvées par les autres membres du groupe de la communauté. En d'autres termes, le projet conçu devra être capable de garder la trace des requêtes effectuées par un membre d'un groupe. Ainsi, pour cette requête le logiciel devra être capable de garder en mémoire quels ont été les liens les plus « visités », nous obtiendrons donc la liste des liens, retournée par le moteur de recherche, triée du plus « visité » au moins utilisé. Ainsi, si un autre membre de ce groupe effectue la même recherche (ou une recherche similaire), la trace des utilisateurs ayant soumise auparavant cette requête sera consultée et une liste de liens sera retournée par rapport aux différentes traces.

Finalement, l'objet de ce projet est de développer un moteur orienté communauté qui profite des expériences passées du groupe afin de modifier le tri des résultats retournés par les moteurs existants. Ainsi, le moteur proposé agira comme un Méta-Moteur de Recherche.

c) Type d'utilisateur

L'utilisation du Méta-Moteur ne requerra pas un niveau élevé de connaissances. En effet, l'utilisateur devra juste installer un agent. Ensuite, il n'aura plus qu'à effectuer ses recherches par le biais d'un formulaire. Les réponses trouvées seront retournées de la même manière que le fait un moteur de recherche, avec en plus les liens surlignés pour ceux dont l'ordre a été modifié. Par conséquent, le logiciel répondra aux attentes des différentes catégories d'utilisateurs.

d) Contexte d'utilisation

Ce projet aboutira à un Méta-Moteur de recherche orienté communauté. Il sera sollicité à chaque fois qu'un utilisateur souhaitera effectuer une recherche sur le Web. Ainsi, au lieu de se servir directement du moteur de recherche en question, il passera par ce Méta-Moteur afin d'avoir des résultats triés.

Le Méta-Moteur nécessite une phase d'initialisation durant laquelle certaines fonctionnalités pourront sembler limitées. Cette phase sera communément appelée phase d'apprentissage.

Pour une première utilisation, l'utilisateur devra installer un agent sur son ordinateur. Ainsi, c'est cet agent qui interagira avec les autres agents connectés. Par la suite, il suffira juste d'entrer l'URL de ce Méta-Moteur dans la barre d'adresse d'un quelconque navigateur Web et d'y effectuer ses recherches. Son utilisation est similaire à un quelconque moteur de recherche. La seule différence, par rapport à un moteur de recherche classique, est la possibilité d'installer un agent pour augmenter la pertinence des résultats.

2) Analyse de l'existant

Le projet est destiné à être multi-plateforme, c'est dans cette optique qu'il a été décidé d'être implémenté en JAVA. Le logiciel est destiné, principalement, à être utilisé sous Linux. C'est dans cette perspective que nous développerons le logiciel sous ce même système d'exploitation et de même pour effectuer les tests de validation.

Pour l'environnement fonctionnel du logiciel, il suffira d'installer la machine virtuelle JAVA, et ensuite d'installer l'agent. Il faudra également penser à affecter un port libre de communication pour l'agent sur le poste utilisateur. Cette installation fera l'objet d'une documentation utilisateur.

De plus, aucune formation pour le client ne sera nécessaire.

II/ Réponse aux besoins

1) Fonctionnalités

Ce projet peut être divisé en plusieurs fonctionnalités définissant ainsi les différents modules à développer.

- Un serveur HTTP/1.1 compatible avec la RFC 2616 mais n'offrant que les fonctionnalités suivantes :
 - Méthode HEAD,
 - Méthode GET,
 - Méthode POST,
 - Implémentation des requêtes en parallèle (thread),
 - Enregistrement des connexions pour une administration plus aisée sur fichier texte et BDD,
 - Paramètre de connexion sur un fichier de configuration en texte ASCII.
- Un module d'interrogation de moteurs de recherches :
 - Avec la possibilité d'effectuer une recherche sur plusieurs moteurs en même temps.
- Un moteur de classement des listes de résultats reçu par les moteurs de recherches.
- Un moteur de classement des listes de résultats par les autres agents suivant l'expérience de recherche des autres utilisateurs (algorithme de tri coopératif fourni par le client).
- Un module d'interrogation des autres agents permettant d'envoyer une liste triée de résultats et recevant la même liste triée différemment
 - Requête POST sur des serveurs HTTP/1.1 distant,
 - Export au format XML de liste triés de résultats à la fois pour la requête que pour la réponse serveur.
- Une base de données répertoriant :
 - Les mots clés recherchés,
 - Les listes d'URL triées par les moteurs de recherches,
 - Les mots-clés associés aux URL (description du lien),
 - Les autres agents à interroger (IP et port),
 - L'identificateur de chaque agent. (Remarque : l'idée est de pouvoir lancer un même agent à partir de différentes adresses IP).
- Une interface graphique ergonomique :
 - Une page de recherche (avec sélection de moteurs de recherches à interroger),
 - Une page d'affichage de résultats (avec surlignage des liens dont l'ordre a été modifié).
- Un serveur enregistrant les agents en ligne (aussi appelé « *registry* » ou registrant).
- Le « *registry* » ou registrant sera chargé de centraliser les (IP, port) des agents connectés et à la connexion l'agent pourra demander cette liste.
Le registrant aura en sa possession tous les identifiants des agents connectés.
Chaque agent sera identifié par son identifiant. Ce dernier est très important dans la mesure où chaque ordinateur ne possède pas forcément un IP fixe. C'est dans cette perspective que chaque agent, installé sur un ordinateur, doit avoir un identifiant unique

Note : Selon l'avancement du projet il se pourrait que le serveur n'implémente pas toutes les fonctionnalités énumérées ici pour des raisons de respect de délais.

Notamment, le serveur registrant sera remplacé par une liste d'agents à contacter enregistré dans le fichier de configuration. Le serveur HTTP pourra se limiter dans un premier lieu à une compatibilité HTTP/1.0 (non-implémentation de la requête HEAD). Nous n'implémenterons au départ que la recherche sur un moteur de recherche : Google (mais la possibilité d'inclure d'autres moteurs de recherches est envisagée). Nous nous limiterons aussi à un tri sur 20 résultats (la possibilité d'étendre cette liste est incluse aussi).

Ces limitations sur le projet son en accord avec le client. Cependant une attention particulière est à faire lors de la conception afin de rendre le prototype facilement extensible pour y inclure de nouveaux services ou pour changer la mise en oeuvre des services existants (i.e. modification de l'algorithme de tri, ajout de support d'autres moteurs de recherches).

2) Préconisations techniques

a) Architecture globale et intégration au SI client

Notre projet a une architecture globale: Modèle, Vue, Contrôleur. Cette architecture permet de séparer le code en trois parties distinctes :

- La vue correspond à l'interface graphique (feuilles de style CSS),
- Le modèle correspond à nos données (base de données),
- Le contrôleur fait le lien entre les deux entités précédentes.

Cependant, notre projet intègre en son sein à la fois un client et un serveur HTTP. Nous adopterons pour cette partie une architecture client/serveur. Ce découpage architectural du projet facilitera ainsi sa modulation.

En ce qui concerne les flux d'informations, notre agent devra pouvoir communiquer avec un client par l'intermédiaire d'un navigateur Web mais aussi, il devra communiquer avec Internet par le biais de requêtes. Un agent devra également communiquer avec d'autres agents par le biais de requêtes XML.

b) Techniques mise en oeuvre

- Serveur HTTP en JAVA avec les classes libres disponibles sous GNU/Linux.
- Moteur de classement de liste de façon coopérative en JAVA avec les classes libres disponibles sous GNU/Linux également (le fonctionnement de l'algorithme est décrit dans une documentation fournie par le client).
- Interface graphique de recherche et de réponse aux clients ergonomique en XHTML 1.0 Strict et CSS2 valide aux standards du W3C.
- Export de listes de résultat triée en XML.
- Système de gestion de base de données choisit : MySQL 4.1.x minimum (<http://www.mysql.com>)
 - Gratuit pour un usage non commercial,
 - Multi-plateforme (disponible sous GNU/Linux)

c) Exigences opérationnelles

En ce qui concerne le temps de réponse, nous prendrons en compte le time-out inter-agents (délai maximum de réponse acceptable). En effet, notre agent doit, dans un premier temps, communiquer avec le Web (temps négligeable), puis doit interagir avec les autres agents en ligne afin d'obtenir une liste de résultats triés (temps variable). Nous estimons que cela ne gênera pas l'obtention

de la liste triée demandée. Les problèmes de connexion inter-agents ne devront pas impacter les performances (temps réponses).

Pour la gestion d'accès, ce projet n'intègre pas de modules d'authentification. Un client voulant utiliser le Méta-Moteur devra au préalable installer un agent sur son ordinateur.

Nous avons également prévu pour l'administration de l'agent de consigner toutes les informations dans un fichier log. Ce fichier nous permettra d'une part de vérifier le bon fonctionnement de l'agent et d'autre part de traiter les erreurs éventuelles.

En outre, il est important de préciser que le protocole de communication inter-agents doit vérifier certaines propriétés telles que :

- Le non-blocage,
- Le fait qu'il n'y ait pas de famine.

3) Investissement matériel et logiciel

L'implémentation du Méta-Moteur ne nécessite l'achat d'aucun matériel et logiciel informatique. Diverses ressources sont quant à elles mises à disposition des développeurs et du client (pour vérifier l'avancement du projet).

Ressources mises à disposition :

- Une liste de diffusion,
- Un wiki (site Internet collaboratif regroupant toutes les informations disponibles sur le projet),
- Un CVS (outil de gestion de code source communautaire),
- La messagerie instantanée

4) Impacts organisationnels

L'utilisation du Méta-Moteur ne nécessite aucune formation. Il sera néanmoins accompagné d'une documentation technique et utilisateur. La documentation technique est destinée à la description algorithmique du logiciel, utile pour les analystes et les programmeurs. Tandis que la documentation utilisateur décrira l'installation et l'utilisation du Méta-Moteur.

5) Moyens fournisseurs

a) Moyens matériels et logiciels

Un article traitant d'« Un système coopératif pour le tri des résultats de moteurs de recherche » est mis à disposition par le client pour aider le groupe à l'analyse et à l'implémentation du Méta-Moteur.

b) Moyens humains

Aucun profil type défini.

III/ Organisation

Compétences requises :

4 développeurs dont 1 chef de projet.

Phases en fonction du temps :

- Analyse	15%
- Conception	40%
- Implémentation	35 %
- Tests	10 %

Nous estimons que chaque personne du projet s'investira 1,5 jour par semaine du 13 février au 22 juin 2006, soit 19 semaines, équivalant à 114 jours/homme de travail.

Fonctionnalités :

- Tri coopératif de résultats inter-agents	40 Jours/homme
- Récupération des liens par les moteurs de recherche	15 Jours/homme
- Serveur HTTP	28 Jours/homme
- Client HTTP	15 Jours/homme
- BDD	10 Jours/homme
- Module XML (spécification de solution)	4 Jours/homme
- Interface Graphique	2 Jours/homme

Cette répartition en Jour/homme par rapport aux fonctionnalités du logiciel pourra être ré-étalonné suivant l'avancement du projet en respectant bien entendu la date de rendu du programme fixé le 22 juin 2002.