Un moteur de recherche multilingue (choix, paramétrage et industrialisation)

Yannick Le Ny

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

PARIS

MEMOIRE

présenté en vue d'obtenir

le DIPLOME D'INGENIEUR CNAM

SPECIALITE : Informatique

OPTION : Systèmes d'information

par

Yannick LE NY

Un moteur de recherche multilingue

(Choix, paramétrage et industrialisation)

Soutenu le 3 juillet 2013 (date définitive)

JURY

PRESIDENT :	**M. le professeur Jacky AKOKA**	**(CNAM Paris)**
ENCADRANT :	**Mme le professeur Isabelle WATTIAU**	**(CNAM Paris)**
MEMBRES :	**Mr le professeur Fayçal HAMDI**	**(CNAM Paris)**
	Mr Ludovic LE GAL	**(EDF)**
	Mr Florian SCHNEIDER	**(EDF)**

Remerciements

Tout d'abord, pour m'avoir permis de réaliser ce mémoire qui va me permettre de terminer mon long cycle d'études en informatique au CNAM, je tiens à remercier Mr Ludovic LE GAL. Je n'oublie pas qu'il a fait le nécessaire pour que je puisse avoir un stage à EDF, ce dernier me permettant de réaliser ce mémoire et de sa patience lors de la rédaction de mon mémoire.

Je remercie Mme Isabelle WATTIAU de m'avoir accepté pour la rédaction de mon mémoire puis de m'avoir guidé pour la rédaction de celui-ci, mais je dois aussi dire que j'apprécie beaucoup sa patience et de ses nombreuses remarques qui m'ont permis approfondir ce mémoire.

Je remercie tous les professeurs du CNAM que j'ai côtoyé au cours de mon cursus en informatique.

Liste des abréviations

API : Application Programming Interface

ASCII : American Standard Code for Information Interchange

ASP : Active Server Pages

CD-ROM : Compact Disc - Read Only Memory

CEI : Communautés des États Indépendants

CGI : Common Gateway Interface

CMMi : Capability Maturity Model + Integration

CobiT : Control Objectives for Information and related Technology

CP : Code Page

CSP-IT : Centre de Service Partagé-Informatique et Télécom

CSS : Cascading Style Sheets

DIT : Direction Information et Télécoms

DMZ : DeMilitarized Zone

DSP : Direction des services Partagés

EDF : Electricité De France

ESI : Exploitation des Services Internet

ESA : Expertise des Serveurs d'applications

FTP : File Transfer Protocol

GDF : Gaz De France

GFDL : GNU Free Documentation License

GNU : GNU's Not UNIX

GNU-GPL : GNU General Public Licence

HDFS : Hadoop Distributed File System

HTML : Hypertext Markup Language

HTTP : HyperText Transfer Protocol

HTTPS : HyperText Transfer Protocol Secure

ISO : International Organization for Standardization

ITIL : Information Technology Infrastructure Library

JSP : JavaServer Pages

JVM : Java Virtual Machine

LDAP : Lightweight Directory Access Protocol

MIME : Multipurpose Internet Mail Extensions

MP3 : MPEG-1/2 Audio Layer 3

NFS : Network File System

NNTP : Network News Transfer Protocol

ODBC : Open Database Connectivity

ODF : Open Document Format

PDF : Portable Document Format

PERT : Program ou Project Evaluation and Review Technique

PHP : Hypertext Preprocessor

PS : Postscript

QSOS : méthode de Qualification et de Sélection de logiciels Open Source

RSS : Really Simple Syndication

RTF : Rich Text Format

SGBD : Système de Gestion de Base de Données

SGML : Standard Generalized Markup Language

SLA : Service Level Agreement

SLO : Service Level Objectives

SQL : Structured Query Language

SSL : Secure Sockets Layer

TRA : Tierce Recette Applicative

TTF : TrueType Font

UCS : Universal Character Set

UML Unified Modeling Language

UTF : UCS Transformation Format

URL : Uniform Resource Locator

XML : Extensible Markup Language

Table des matières

I Introduction

Dans ce chapitre, je vais détailler la nécessité d'un moteur de recherche pour les utilisateurs, puis je détaillerai le besoin au sein d'EDF dans le cadre de mon mémoire.

I.1 Le besoin de moteur de recherche pour des sites Internet et Intranet

Les utilisateurs ont besoin d'un moteur au quotidien pour différentes raisons que je vais détailler ci-dessous.

I.1.1 Recherche par le contenu

La notion de recherche par le contenu veut dire qu'on peut retrouver un document grâce à son contenu, et non par un identifiant quelconque comme c'est le cas pour les bases de données. On peut donner comme exemple celui de retrouver un jugement entre 2 parties dans les greffes d'un tribunal, puisqu'ici une utilisation de la recherche par contenu est nécessaire. Cette nécessité est dictée par le fait qu'on ne peut citer aucun identifiant unique dans les termes de la recherche comme le numéro 78-17 pour la loi « Informatique et libertés ». Donc avec ce type de recherche, on peut potentiellement avoir des résultats avec une connaissance préalable relative à la présence de termes approchants dans un ou des documents.

I.1.2 Information « non structurée » opposée à information « structurée »

En plus de la différence qu'il faut faire entre recherche par contenu via un moteur de recherche dans un index ou recherche par identifiant unique dans une base de donnée via un connecteur, il y a aussi 2 types d'information. L'une est dite « structurée » et se trouve habituellement dans des bases de données relationnelles dans lesquelles les entreprises ont entre 20 et 40% de leur information « opérationnelle ». L'autre est dans les 60 à 80% restant et est constituée de textes stockés dans des fichiers semi-structurés voire non structurés. Les documents provenant de suites bureautiques (traitement de texte,

tableur, logiciel de présentation), de messageries, de fichiers PDF ou de pages web forment l'ensemble des documents semi ou non structurés (à la différence des bases de données) qui font partie de la mémoire de l'entreprise. Les rédacteurs de ces documents les déposent suivant leur volonté sur des partitions de disques durs ou de partages réseaux (système de fichiers), sur des serveurs web ou encore dans des applications de collecticiels (« groupware ») qui permettent un travail en mode collaboratif comme les bases Lotus Notes.

I.1.3 Nécessité d'un moteur de recherche sur les sites web

Augmentation du nombre de documents et du multilinguisme sur Internet

Le nombre de documents sur Internet est en augmentation de manière régulière voire exponentielle depuis le milieu des années 90. D'après une étude de 2001, il y avait globalement plus de 550 milliards de documents sur le Web, principalement dans le Web invisible, ou Web profond [DEEP-WEB]. Le 25 Juillet 2008, les ingénieurs logiciel Jesse Alpert et Nissan Hajaj de Google ont annoncé que Google Search avait découvert 1000 milliards d'URL uniques [WEB2008]. En Mai 2010, d'après une évaluation faite avec plusieurs moteurs de recherche, le Web indexable contenait au moins 21,54 milliards de pages [WEB2010]. Au 31 Mai 2010, il y avait plus de 119,5 millions de sites web en fonctionnement [DOMAIN2010] dont 73% étaient des sites commerciaux ou autres opérant avec la racine du domaine en .com. On voit donc d'après toutes ces études que le nombre de pages sur Internet ne fait que croître.

Il en est de même au sein de l'entreprise où se retrouve cette « explosion des données informatisées » qui aboutit à une augmentation croissante de l'espace de stockage et du nombre de serveurs d'applications chargés de délivrer cette information aux utilisateurs. Les entreprises dont l'activité est basée sur la vente ou la gestion de l'information ne font que créer, traiter et stocker des milliards de fichiers ou d'enregistrements dans des bases de données sur des serveurs situés au niveau national ou international. Chaque jour, sans cesse, de nouvelles données sont ajoutées alors que d'autres sont modifiées ou supprimées. Par exemple dans les sites de la presse en ligne, de manière quotidienne, de nouveaux articles sont ajoutés et de temps en temps modifiés tandis que les lecteurs commentent ces mêmes articles.

Afin de pouvoir utiliser cette information dont le volume croît sans cesse, les employés des entreprises ou les internautes doivent avoir à disposition des outils de recherche performants. La plupart du temps en entreprise c'est à l'utilisateur de faire lui-même le tri dans le vaste ensemble des sources de documents mis à sa disposition. Pour améliorer sa productivité, il est donc nécessaire que son temps passé à la recherche soit le plus faible possible pour récupérer les documents dont il a besoin dans le cadre de son travail. Dans le cadre d'un site web de commerce en ligne, il est indispensable que le moteur de recherche du site puisse lui fournir des résultats à partir d'une requête simple. Dans le cas contraire, il se tournera vers un site qui lui fournira une réponse pour le produit qu'il souhaite acheter. Chaque requête de recherche qui n'aboutit pas alors que le produit est disponible est potentiellement une vente perdue.

Une enquête de 2002 portant sur 2 024 millions de pages Web a déterminé que la plupart du contenu du Web était en anglais: 56,4%; ensuite viennent les pages en allemand (7,7%), français (5,6%) et japonais (4,9%) [WEB2002]. Une étude plus récente, qui a utilisé des recherches sur le Web dans 75 langues différentes pour échantillonner le Web, a déterminé qu'il y avait plus de 11,5 milliards de pages Web dans le Web public indexables à la date de fin Janvier 2005 [WEB2005]. Ces derniers renseignements peuvent être complétés par une autre étude de fin 2009 qui ne concerne pas les langues des pages web mais celles des Internautes au niveau mondial, où l'on a donc : 27.7 % ont pour langue maternelle l'anglais, 22.6 % le chinois, 7.8 % l'espagnol, 5.3 % le japonais, 4.3 % le portugais, 4.0 % l'allemand, 3.3 % l'arabe et 3.2 % le français. On voit donc qu'en l'espace de 7 ans, les utilisateurs parlant l'espagnol, le portugais ou l'arabe sont plus présents sur la toile mais ce n'est rien comparé au raz de marée chinois qui est, de plus, tout à fait normal eu égard à leur pourcentage dans la population mondiale. On voit donc, d'après toutes ces études, que l'Internet devient de plus en plus multilingue et que le règne de langue anglaise du milieu des années 90 est de plus en plus remis en cause au fil du temps. Il en est de même dans les entreprises qui sont de plus en plus internationales et qui utilisent habituellement dans leurs documents la langue du pays où l'employé se trouve et la « lingua franca » internationale depuis la seconde guerre mondiale qui est l'anglais. Les systèmes de recherche doivent prendre en compte ces contraintes de multi-linguisme et d'encodage informatique des documents avec des jeux de caractères parfois très différents comme dans le cas de l'écriture de langues asiatiques.

Les moteurs de recherche Internet (durée d'utilisation et porte d'accès sur Internet)

En Mars 1998, une étude de Media Metrix / PC Meter donnait un temps mensuel passé sur les moteurs de recherche suivants : 22,3 mn sur Yahoo, 12,8 mn sur Excite et 10,7 mn sur le moteur Altavista que j'utilisais alors [MDR98]. En Juillet 2004, une étude de Nielsen donnait : 29mn57s pour Google, 28mn28s pour Aol Search et 13mn09s pour Netscape [MDR04]. En mars 2010, une autre étude Nielsen indiquait que les utilisateurs passaient en moyenne par mois sur les sites des marques suivantes : 1h18 pour Google et 2h35 pour Yahoo (l'activité de recherche pure hors actualités ou autres n'est pas précisée) [MDR10]. On voit donc que la durée d'utilisation des moteurs de recherche progresse. Ceci est en corrélation avec le temps passé sur Internet par mois par les Internautes qui a augmenté en moyenne de 6h13mn en Octobre 2002 à 24h12mn en Octobre 2006 [TEMPS]. De plus la recherche se fait de plus en plus par moteur de recherche puisqu'en Juillet 1997, une étude CommerceNet/Nielsen Media indiquait que l'on arrivait sur un site à 71% des cas par un moteur de recherche [MDR98]. Ce pourcentage est passé à 81% en 2000 d'après UK Internet User Monitor - Forrester Research [MDRSTAT].

Le moteur de recherche est donc la porte presque obligée pour l'accès à l'ensemble des sites web de l'Internet.

Les attentes des utilisateurs des moteurs de recherche

La société iProspect [iprospect] a réalisé une étude très intéressante au printemps 2004 qui montre les comportements des utilisateurs face aux moteurs de recherche web.

56 % des personnes disaient qu'ils utilisaient un moteur de recherche au moins une fois par jour. Près de 82 % des utilisateurs indiquent que, s'il n'y a pas des résultats qui leur conviennent dans les 3 premières pages, alors ils font une nouvelle recherche. Les professionnels du bâtiment, plus que toute autre profession, s'arrêtent aux résultats sur la première page. Les personnes âgées ont le même comportement que ces derniers. Les utilisateurs de moteurs de recherche trouvent que les résultats de recherche en langage naturel (organiques ou algorithmiques) correspondent le plus à leur recherche, c'est à dire en utilisant une requête de recherche avec une phrase et non une suite de mots-clés.

Une étude de la société NPD [MDRSTAT] a étudié comment les utilisateurs recherchaient de l'information. Les résultats montrent que 45% des utilisateurs recherchent en utilisant de multiples mots-clés ou phrases clés, 28% utilisent un mot-clé, 18% recherchent avec une option pré-définie (tel que naviguer dans une catégorie d'un annuaire) et 9% recherchent en tapant une question.

On voit donc qu'il est important d'avoir des résultats pertinents dès les toutes premières pages de résultats avec un et surtout plusieurs mots-clés. Dans l'entreprise, l'optimisation des résultats de recherche en fonction de mots-clés spécifiques au métier de l'entreprise peut-être nécessaire afin d'augmenter la productivité des employés, surtout pour les employés réalisant des travaux intellectuels et ne faisant que du traitement de l'information à longueur de journées.

I.2 Présentation

La partie ci-dessous présentera succinctement le contexte de l'entreprise puis le besoin d'un moteur de recherche à EDF.

I.2.1 L'entreprise

Ce mémoire s'appuie sur une étude réalisée de 2005 à 2006 en partie au sein de l'équipe du service ESI (Exploitation des Services Internet) chargé de l'hébergement (exploitation et administration) de plusieurs centaines de sites web d'EDF (Electricité De France) – GDF (Gaz De France). Ce service appartenait à la DIT (Direction Information et Télécoms) devenue depuis DSP (Direction des services Partagés)/CSP-IT (Centre de Service Partagé-Informatique et Télécom). Le service ESI n'existe plus à l'heure actuelle suite à la séparation totale des services informatiques communs à EDF-GDF au 01/01/2010. Puis la réflexion a continué au sein du service ESA (Expertise des Serveurs d'applications) au cours des années suivantes.

I.2.2 Le projet

Le système de recherche existant

A la fin des années 90, le moteur de recherche Verity Search'97 a été installé sur 2 serveurs dont l'un sur l'Intranet et l'autre sur l'Internet. Ces 2 serveurs avaient pour objectif d'offrir un service de recherche aux sites web d'Electricité De France et de Gaz De France hébergés par l'entité ESI. Au début de l'année 2003, un nouveau moteur de recherche utilisant Verity K2 4.5 a été mis en place afin d'offrir un nouveau service de recherche utilisable par l'ensemble des sites Intranet d'EDF et mixtes EDF-GDF.

Le coût très important des licences, de l'hébergement et de l'exploitation a été pris en charge par la Direction de la Communication d'EDF afin de faciliter la recherche restreinte sur chacun des sites web ou une recherche globale sur tout l'intranet d'EDF à partir du site portail de l'intranet d'EDF appelé E-toile. Ce moteur a aussi permis d'indexer des données qui ne l'étaient pas auparavant et qui étaient stockées dans des bases Oracle, des bases Lotus Notes, des bases documentaires Documentum tout en gérant les autorisations grâce aux listes de contrôle d'accès de ces logiciels.

La mise en place d'un serveur Verity K2 sur la zone Internet en zone démilitarisée (DMZ) afin de permettre d'indexer les sites Internet d'EDF a été envisagée mais n'a pas été effectuée pour des questions financières. De plus l'utilisation du Verity K2 en Intranet pour indexer les sites Internet aurait été possible pour les indexations uniquement, mais l'utilisation pour la recherche n'aurait pas été possible à cause d'un flux descendant de la zone réseau Internet vers la zone réseau Intranet. Or ce type de flux est strictement interdit par l'équipe réseau pour des questions de sécurité.

Verity Search'97 en Internet est utilisé pour indexer des sites en anglais et en français. Ces sites peuvent aussi être en HTTP et HTTPS et être non sécurisés ou sécurisés via un identifiant et un mot de passe.

En 2005, le moteur Verity Search'97 fonctionnait sur un serveur utilisant le système d'exploitation Sun Solaris 6. Or il y avait un risque en cas de migration forcée sur un serveur avec une version plus récente de Solaris, pour cause de récupération de place dans la grande salle serveur, alors que cette version du système d'exploitation n'était pas indiquée comme supportée par la dernière version de la documentation de Verity. De plus l'installation du logiciel n'était plus possible puisque le CD-ROM d'installation semblait

avoir été égaré. A ces problématiques, Gaz de France dont nous hébergions le site institutionnel www.gazdefrance.com était demandeur d'un nouveau moteur de recherche dans plusieurs langues européennes qui n'étaient pas l'anglais et le français.

Lors de cette même année, en tant qu'exploitant, j'étais confronté à la demande du projet Portail Salarié Groupe d'EDF qui demandait un moteur de recherche multi-lingue (français, anglais, allemand). Or ce projet n'était pas prêt à investir dans le moteur de recherche Verity K2 et ne pouvait exploiter le moteur Search'97 de l'Internet ou l'Intranet parce qu'il était dans une zone réseau spéciale.

Les limites techniques et fonctionnelles du service de recherche existant

Aujourd'hui, ce moteur de recherche ne répond plus au besoin sur au moins sept aspects essentiels :

- le support : le produit n'est plus supporté par la société Verity. De plus le CD-ROM d'installation et les justificatifs de numéro de licence ont été égarés,
- la sécurité : Search'97 n'est pas capable d'indexer les sites sécurisés par SSL en HTTPS, alors que nous avons un certain nombre de sites de ce type,
- l'identité visuelle : celle-ci n'est pas en accord avec les chartes graphiques en vigueur,
- la pertinence : Search'97 n'est pas en mesure de trouver des résultats avec et sans accents suite à la saisie d'un mot-clé,
- l'ergonomie: il n'y a pas de surbrillance sur le ou les mots-clés recherchés dans les résultats de la recherche,
- les langues : L'instance Search'97 en Internet ne supporte que l'anglais et le français,
- les documents bureautiques : Les parseurs des documents ne sont plus à jour par rapport aux nouveaux types de document qui utilisent la norme XML pour le stockage et la structuration.

Objectif du prototype et du mémoire

Le prototype « Moteur de recherche multilingue » réalisé dans le cadre du mémoire résulte de l'industrialisation d'un logiciel de moteur de recherche. Ce logiciel est destiné à être soumis à validation de la part du comité d'architecture technique en vue de le référencer pour s'intégrer au référentiel technique d'EDF. Ce référencement permet le support et le libre déploiement au sein de l'entreprise. Dans le cas contraire une dérogation est nécessaire lors de la validation du dossier d'architecture d'une application.

Les attentes exprimées par le client pour le futur moteur de recherche sont diverses.

On peut citer en premier les besoins essentiels. Actuellement, les sites utilisent de plus en plus l'authentification via SSL sur HTTP au sein d'EDF. De plus les sites internet qui sont hébergés au sein de la DSP/CSP-IT sont de plus en plus multilingues : langues d'Europe centrale, occidentale à cette date et celles de la CEI (Communautés des États Indépendants, ex-URSS sans les pays baltes) dans le futur. Et la mise en place de la recherche au sein d'un site doit être la plus facile à mettre en œuvre via des éléments standards comme une page de recherche habillée à la charte Internet d'EDF.

Ensuite, il est fortement nécessaire que les recherches effectuées avec des mot-clés sans les accents habituels rapportent les résultats contenant le ou les mots-clés dans les documents avec ou sans l'accentuation. Un affichage des recherches clair et fonctionnel avec une navigation facile est requis. Les formats de documents de la suite bureautique Microsoft Office devront être indexés.

Les fonctionnalités optionnelles seront précisées ci-dessous. La mise en surbrillance ou en relief d'une manière graphique quelconque du mot-clé dans les résultats de la recherche et dans des copies en cache comme dans Google est fortement souhaitée. L'indexation de documents compressés au format zip serait utile. Ensuite une recherche via les opérateurs booléens permettrait de satisfaire les utilisateurs des sites web les plus exigeants.

L'industrialisation de ce moteur de recherche a été le cadre qui a permis la réalisation de ce mémoire. Ce dernier, au-delà de la réalisation technique, permet de montrer la démarche de choix puis d'industrialisation d'un moteur de recherche multilingue afin de l'intégrer au sein du système d'information d'EDF. Ce mémoire décrit les différents aspects de la mise en place de ce service de recherche : aspects techniques et aspects fonctionnels.

Un cahier des charges pour le choix des fonctions offertes par le moteur a été rédigé dans un premier temps puis a été soumis pour validation au tuteur de ce mémoire. Ce cahier des charges fait l'objet d'une documentation EDF appelée « Cahier des charges du moteur de recherche Open Source ». Le choix du moteur de recherche retenu, mnoGoSearch, est le résultat d'une étude de marché présentée au chapitre 3 qui sera suivie respectivement des fonctionnalités des moteur de recherche dont l'aspect multilingue, de son industrialisation et des aspects techniques.

I.2.3 Le document

Ce mémoire est présenté dans l'objectif d'obtenir le diplôme d'ingénieur en informatique du Conservatoire National des Arts et Métiers dans la poursuite du cursus « Informatique option Ingénierie des Systèmes d'Information ». La rédaction et l'industrialisation du logiciel ont été réalisées dans le cadre d'un travail initié en 2005-2006. Comme le mémoire n'a pas été soutenu à ce moment là pour des raisons diverses, il a été complètement réactualisé depuis la fin 2010 sous la direction du Professeur Isabelle Wattiau.

Les chapitres du mémoire sont dans l'ordre suivant :

- Le chapitre 2 est une présentation générale des moteurs de recherche. Il permet d'initier le lecteur aux concepts de base.
- Le chapitre 3 informe sur la méthode d'évaluation des logiciels Open Source retenus puis détaille l'étude des logiciels choisis afin de retenir le logiciel correspondant le plus au cahier des charges.
- Le chapitre 4 décrit d'un point de vue fonctionnel le moteur de recherche.
- Le chapitre 5 expose les fonctionnalités attendues sur un moteur de recherche multilingue.
- Le chapitre 6 décrit les aspects techniques du moteur de recherche choisi.
- Le chapitre 7 informe des enseignements tirés de cette étude et décrit l'industrialisation du moteur
- Le chapitre 8 regroupe les annexes.

II Les moteurs de recherche (Utilisation et Fonctionnement)

II.1 Principe de fonctionnement d'un moteur de recherche

Les moteurs de recherche ont pour objectif de faciliter l'accès à différents sources d'information en les indexant puis en permettant à l'utilisateur de trouver des documents via une interface de recherche. Les moteurs de recherche sont constitués de 5 parties que nous allons détailler : « spider », indexation, classification, recherche, présentation des résultats.

II.2 Le spider

Le « spider » est un logiciel qui a pour objectif de récupérer les documents de différents types sur une source de données distantes. Pour effectuer ce travail, le moteur de recherche doit lui transmettre la source de données sous forme d'URL comme http:// , ftp:// ou file:// et lui indiquer les URLs ou extensions de fichiers qui sont exclues à la récupération. Ce logiciel nécessite donc un support des principaux protocoles de communication Internet et d'autres sources de données comme les serveurs de bases de données. De plus il doit implémenter des fonctions d'identification de l'encodage du document.

Un moteur de recherche a besoin d'accéder aux données distantes via une partie logicielle spécifique. Dans le cas des bases de données utilisant la norme SQL (sigle de Structured Query Language) ou des bases de messagerie Lotus Notes ou MS Exchange, il s'agit d'un simple connecteur utilisant les API (Application Programming Interface) des librairies des clients proposées par ces fournisseurs.

Dans le cas de l'accès à des sites Internet, le moteur de recherche web implémente une partie spécifique au moteur qui supporte les principaux protocoles Web : HTTP et HTTPS (serveur web), FTP (serveur de fichier) , NNTP (serveur de nouvelles). Grâce à cette connectivité réseau, le spider peut alors explorer les données du serveur distant en utilisant un algorithme, qui, à partir d'une ou plusieurs URL de départ, permet de lister l'ensemble des documents hébergés sur le serveur.

Après qu'il ait reçu un ensemble d'URLs racines pour commencer la recherche, il suit tous les liens sur ces pages de manière récursive pour trouver toutes les nouvelles pages.

Des mécanismes permettant de détecter qu'une URL a déjà été traitée par le spider sont prévus afin de s'assurer que les documents ne sont pas récupérés plus d'une fois.

Il existe de plus une norme de courtoisie qui indique au moteur si certaines parties du site ne sont pas à indexer par l'intermédiaire d'un fichier appelé robots.txt déposé à la racine du site. La plupart des moteurs de recherche respectent cette règle.

Les pages web et documents bureautiques, trouvés par le spider sur le web, seront alors d'abord rapatriés en local puis indexés. L'indexation ne peut se faire que lorsque les données des différentes sources sont en local. En même temps que la récupération des documents web, le spider associe des informations comme la date de création ou le type MIME (Multipurpose Internet Mail Extensions) (*.doc, *.xls, *.odt, *.txt, etc …) qui sont fournies par le serveur avec chacun des documents.

II.3 Principe d'indexation du texte

Après avoir été rapatriés en local, les pages web et les documents web vont pouvoir être indexés c'est-à-dire que l'ensemble des mots d'un document vont être analysés et stockés avec différentes informations contextuelles dans un fichier d'index afin d'obtenir une recherche rapide. Habituellement les données de cet index sont stockées sous la forme d'un index inversé. Après récupération du document, l'étape globale dite d'indexation effectuera un parsing du texte en ne tenant pas compte de la mise en page puis différentes étapes de traitement seront appliquées sur ce texte : suppression des mots vides, lemmatisation, identification d'idéogrammes formant un mot dans des écritures comme le chinois, etc...

II.3.1 Les mots-clés

Les mots-clés sont des mots qui portent un sens et seront utilisés par les utilisateurs pour faire des recherches. On peut les opposer aux mots-vides qui servent de liaison dans les phrases mais ne portent pas de sens.

II.3.2 Suppression des mots vides

Un texte est constitué d'un ensemble de mots de différents types comme les pronoms (je, tu, il, etc...), des verbes, des noms, des prépositions (à , après, chez, de , etc...), des adjectifs (bleu, noir, grand, petit, etc). Or certains mots ont des fréquences importantes et, de plus, ne sont pas utilisés comme mots-clés pour la recherche par les utilisateurs. L'on retrouve entre autres les pronoms et les prépositions dans cette liste. Afin donc de réduire la taille des index, on utilisera une liste dite de mots vides dont les mots membres ne seront pas indexés. Cette liste se compose souvent d'un fichier texte où un mot est mis par ligne.

II.3.3 Normes d'encodage

L'indexation nécessite de reconnaître les différents mots du texte dans un document. Or ces mots, dont la représentation graphique informatique change suivant les langues, nécessitent un codage informatique appelé page de code. L'anglais, qui utilise l'alphabet latin des langues occidentales, est pauvre en caractères accentués comme l'accent aigu en français ou en caractères spéciaux comme le tilde en espagnol. Mais d'autres langues non latines comme l'arabe, l'hébreu utilisent une graphie spécifique et d'autres comme le chinois n'utilisent pas d'alphabet mais des caractères ayant différent sens en fonction du contexte. Donc l'anglais pourra se contenter de page de code informatique de type ASCII, mais le français ou l'espagnol devront utiliser au minimum l'encodage CP1252 sur MS Windows ou ISO-8859-1 sur Unix/Linux. Mais la tendance est désormais d'utiliser l'encodage au format UTF-8, qui permet d'afficher l'ensemble des langues vivantes au niveau mondial, dans les documents bureautiques et les pages des sites web, malgré le fait qu'il y a parfois des problèmes d'affichage des caractères accentués en français.

II.3.4 Prédéfinition d'un ensemble de caractères formant un mot

Certaines langues n'utilisent pas des mots avec un alphabet mais avec des caractères ou idéogrammes comme les caractères chinois. Or ces textes rédigés avec des idéogrammes ont la particularité de posséder, pour les textes les plus modernes, une

ponctuation avec un point pour finir la phrase et des virgules mais pas d'espaces entre les idéogrammes permettant de délimiter un mot. Afin d'améliorer par la suite les recherches, il est nécessaire de prévoir un traitement lors de l'indexation et avant le stockage des index qui permet de définir qu'une suite de 2 ou plusieurs caractères forme un même mot et que, lors de la recherche, si un de ces ensembles de 2 ou plusieurs caractères est renseigné par l'utilisateur, alors la recherche portera sur ce sous-ensemble et non sur chacun des caractères distincts. Afin de faire ce traitement lors de l'indexation, on constitue une liste se composant souvent d'un fichier texte où une suite de 2 ou plusieurs caractères constitue une ligne.

II.3.5 Thesaurus

Afin d'améliorer l'indexation, il est possible d'utiliser un thesaurus, ce dernier est une sorte de dictionnaire organisé en concepts par groupe ou domaine, plutôt que par termes. Il contient un ensemble de termes (en général des substantifs) qui peuvent être utilisés pour indexer des textes portant sur un certain domaine : économie, scientifique, sportif, politique, etc.

On distingue dans ce type de dictionnaire :

- les descripteurs qui seront effectivement utilisés pour l'indexation,
- les non descripteurs qui sont des synonymes des précédents.

L'ensemble des termes d'un thésaurus est muni d'une structure de type graphe dont les relations classiques qui sont utilisées sont :

- les relations d'association,
- les relations de synonymie,
- les relations de généricité,
- les relations de spécificité.

II.3.6 La lemmatisation

La lemmatisation est le traitement de l'analyse lexicale du contenu d'un texte regroupant les mots d'une même famille, c'est à dire qu'elle consiste à remplacer un mot

par son lemme. Les mots d'un même contenu textuel se trouvent donc réduits en une entité appelée lemme (sous sa forme canonique). Les différentes formes que peut revêtir un mot comme "le nom, le pluriel, le verbe à l'infinitif, etc" sont regroupées dans la lemmatisation.

Les mots ou lemmes d'une même langue utilisent différentes formes en fonction de leur nombre (un ou plusieurs), leur personne (moi, toi, eux...), leur genre (masculin ou féminin), leur mode (indicatif, impératif...) et donnent alors naissance à plusieurs formes pour un même lemme.

La lemmatisation d'une forme d'un mot consiste donc à en prendre la forme canonique. Cette dernière est définie comme ce qui suit :

- pour un verbe : ce verbe sous sa forme à l'infinitif,
- pour les autres mots (noms) : le mot au masculin singulier.

On peut ainsi noter que toutes les entrées d'un dictionnaire sont lemmatisées. Les mots d'une langue peuvent être classés en deux catégories : les lemmes vus précédemment et les mots obtenus par flexion de ces lemmes : conjugaison d'un verbe, changement de genre ou de nombre, etc.

II.3.7 Les différentes techniques liées à l'indexation

Indexation de type texte intégral

Dans ce mode, qui est la plus simple des techniques, on indexe tous les mots du lexique (noms, verbes, adjectifs, etc) auxquels on retire les mots vides de sens comme les conjonctions de coordination et les pronoms. Puis un fichier d'index est créé suivant une structure donnée et contenant en plus des mots, le document dans lequel ils se trouvent et la ou leurs différentes positions au sein de ce ou ces documents. Cette position peut avoir comme référence de départ le début du document ou un élément de structure de type page ou paragraphe ou section dans le document. Grâce à cette gestion du positionnement, on peut faire des recherches en ayant une requête de recherche avec plusieurs mots dans un certain ordre.

Indexation de type sémantique ou linguistique

Cette méthode permet de détecter les structures sémantiques contenues dans les documents à indexer, cela lui permet de trouver des documents n'ayant en commun aucun terme avec la requête de l'utilisateur. Cette technique permet de répondre aux besoins des utilisateurs dont les requêtes ont pour objectif de trouver les documents qui traitent une idée précise plutôt que différents documents contenants des mots isolés.

Indexation de type statistique

Dans ce mode, les techniques d'indexation se basent sur des statistiques. Dans ce cadre, la méthode aura comme base l'analyse de la présence (ou fréquence) des termes contenus dans la requête de recherche de l'utilisateur et ceux dans les documents. On recherche donc des relations entre la fréquence d'apparition d'un mot dans un document et la pertinence de ce document, et la relation inversement proportionnelle entre l'importance d'un mot-clé et le nombre de documents le contenant. En fait, l'objectif est de privilégier les termes rares suivant le principe, que, plus le terme est rare, plus il sera discriminant et inversement. Avec cette technique, on prend en compte la pondération des termes du document afin de donner un poids aux mots en fonction de leur fréquence d'apparition et de leur position au sein du texte.

II.3.8 Stockage et structuration des index

L'index est un fichier ou une base de données qui sert à stocker les différents termes ou mots porteurs de sens et utilisables comme mot-clés par l'utilisateur lors de sa recherche. Après les différents traitements précédents, les mots clés retenus sont stockés dans un index. Ce stockage se fait en ajoutant des informations de position de chaque mot-clé trouvé dans chaque document et de la liste de l'ensemble des documents, sous forme d'URL source, qui contiennent ce mot-clé.

Les index sont souvent stockés dans un format de fichiers qui utilise la technique des arbres de type « B_tree ». En plus de ce type de technique, il est possible d'utiliser une structure de type séquentiel indexé ou de faire un stockage dans une base de données. Cette structure ordonne les index sous forme de structure hiérarchique. Dès qu'un document est mis à jour, le fichier d'index doit être organisé de manière adéquate afin que le fichier des index soit mis à jour rapidement. Il est à noter que la longueur maximale d'un mot-clé indexé et devant être stocké dans un index est souvent limité entre 25 à 30 caractères dans

le cas du français et que donc ce type de paramétrage doit être pris en ligne de compte lors d'un paramétrage de l'indexeur.

II.4 La classification

En plus de l'indexation et de ses différentes techniques évoquées précédemment, on rajoute un traitement supplémentaire appelé classification dont l'objectif est d'améliorer le temps et la rapidité des réponses. Cette recherche d'information améliorée exploite les associations entre les documents d'une collection. Pour cela, on regroupe les éléments qui se ressemblent en fonction de critères de classification dans des sous-groupes appelés classes. L'utilisateur peut alors obtenir d'autres documents ressemblant au document consulté. On appelle cette opération une hypertextualisation quand les documents concernés sont au format HTML.

II.5 Les différents modes de recherche

Il existe plusieurs modes de recherche qui permettent de formuler et transformer les demandes des utilisateurs en une syntaxe compréhensible par le moteur de recherche, voici les principales :

II.5.1 La Recherche booléenne

Elle est parfois appelée « recherche dite simple » ou « recherche par mots-clés ». Ce type de recherche nécessite l'utilisation des opérateurs booléens : ET, OU, SAUF qui deviennent respectivement en anglais AND, OR, NOT. Pour les utilisateurs avertis, ils peuvent créer des requêtes de recherche compliquées mais qui seront difficiles à déchiffrer pour le « simple mortel ».

Exemple d'une recherche sur les centrales thermiques d'EDF :

((edf) OU (electricite ET de ET france)) ET (centrales ET thermiques)

Ce modèle est très simple pour obtenir un nombre important de documents lors d'une recherche mais au détriment de la qualité de la recherche.

II.5.2 La recherche textuelle

Ce type de recherche utilise des opérateurs particuliers plutôt destinés à des utilisateurs avancés pour faire leurs recherches, parmi ceux-ci on trouve les opérateurs de proximité, le masque et la troncature.

Les opérateurs de proximité :

Ces types d'opérateurs sont destinés à effectuer une recherche très précise en spécifiant les positions attendues de différents mots-clés dans les documents. Les principaux sont :

- la distance : elle permet de demander à ce qu'une distance maximale soit respectée entre 2 mots recherchés,
- l'obligation d'être contenu dans une même partie du texte : cette contrainte s'applique habituellement à une phrase ou à un paragraphe où l'on peut aussi préciser l'ordre des mots-clés,
- l'adjacence : elle permet de rechercher deux mots-clés qui sont l'un à coté de l'autre dans un texte.

Le masque

Ce caractère spécial permet de remplacer un ou aucun caractère. La convention habituelle utilise le caractère dièse « # ».

La troncature

Elle correspond à un caractère spécial, appelé aussi joker, qui remplace une chaine de texte non défini.

II.6 Les modèles de recherche utilisés dans le fonctionnement interne d'un moteur de recherche

Les modèles de recherche sont les différentes méthodes possibles qui sont intégrées dans le moteur et qui permettent de mettre en correspondance les documents recherchés et la requête utilisateur. Voici les principaux modèles :

II.6.1 Le modèle booléen

C'est un modèle à base de logique binaire qui détermine si un document correspond à une requête qui est un ensemble logique de mots-clés reliés par les opérateurs booléens de disjonction (OR), de conjonction (AND) et la négation (NOT). La recherche essaiera de trouver tous les documents correspondant à la cible de la requête utilisant les opérateurs indiqués précédemment qui séparent chaque mots-clé.

II.6.2 Le modèle vectoriel

Ce modèle utilise la relation qui lie l'apparition d'un terme dans un document et l'importance accordée à celui-ci. Pour cela, on utilise un vecteur d'une taille du nombre de mots-clés dans le corpus documentaire qui représente chaque document, et le nombre d'apparitions du mot-clé dans le document qui est représenté par chaque élément de ce vecteur. Pour la recherche, un vecteur représente la requête de l'utilisateur selon le même principe, et ensuite ce vecteur nommé « Vrequête » avec tous les documents contenus dans les index en utilisant la comparaison du cosinus de l'angle entre les deux vecteurs dont un seul est prédéfini.

II.6.3 Le modèle probabiliste

Celui-ci se base sur l'estimation de la probabilité de la pertinence d'un document par rapport à une requête utilisateur et la classification automatique des documents effectués par le moteur de recherche.

II.7 Concept de pertinence de la recherche

Dans le cas de la recherche de documents, il est nécessaire de définir quelques termes qui permettent de préciser ce qui est attendu par l'utilisateur.

- Le **silence** est l'ensemble des documents pertinents mais qui ne sont pas retournés dans la liste des résultats de la recherche.
- Le **bruit** est l'ensemble des documents rapportés dans la liste des résultats de la recherche mais qui ne sont pas pertinents par rapport à la question posée.

Note : Le terme de pertinence utilisé ici représente la pertinence admise par l'utilisateur en fonction des documents trouvés et qui correspondent à son souhait. Il ne s'agit donc pas de la pertinence que le moteur de recherche a pu calculer via des algorithmes à partir de l'équation de recherche.

Dans la théorie, il faudrait minimiser les valeurs du silence et du bruit, mais dans la pratique il s'agit de trouver un compromis qui permet de satisfaire de manière globale l'ensemble des utilisateurs.

En plus des paramètres de silence et de bruit, il en existe 2 autres :

Le rappel = nombre de documents pertinents trouvés / nombre de documents pertinents existants dans l'ensemble des documents

Le rappel sert à mesurer le silence. Lorsque le rappel se rapproche des 100%, il y a moins de silence et la liste des réponses s'améliore.

La précision = nombre de documents pertinents trouvés / total de documents trouvés

La précision sert à mesurer le bruit. Lorsque la précision se rapproche des 100%, il y a moins de bruit et la liste des réponses s'améliore.

Silence = 1 – Rappel **Bruit** = 1- Précision

Et avec un moteur de recherche parfait, on aurait Rappel = Précision = 1

Les 2 indicateurs de Rappel et de Précision permettent de déterminer la performance d'un moteur de recherche, en se basant non sur une seule requête, mais sur des centaines de requêtes différentes.

Dans la pratique, comme ces 2 indicateurs sont opposés, l'utilisateur souhaitera disposer d'un moteur de recherche qui offre un compromis entre ces 2 contraintes :

- ne donner que les résultats d'une recherche qui sont les plus pertinents pour l'utilisateur afin de lui éviter de se « noyer » dans le flot des résultats.
- tendre, dans la majorité des cas, à fournir un résultat non nul à la recherche demandée. Pour certains sites de commerces en lignes, par exemple, le moteur doit toujours proposer un résultat quelle que soit la demande afin de proposer au moins un produit à l'internaute.

II.8 Affichage des résultats

La page de présentation des résultats permet à l'utilisateur de faire sa première sélection de documents qui lui semble correspondre à ce qu'il attend. Les informations sur cette page doivent permettre :

- de comprendre la manière dont le moteur de recherche a effectué la sélection des documents retenus pour la réponse (liste de mots vides, mots choisis, synonymes utilisés de mots de la requête utilisateur, mots trouvés dans ces documents),
- d'avoir une évaluation de la pertinence des documents de la réponse sans nécessité de les consulter tous un par un,
- d'être lues facilement grâce à une mise en page adaptée.

II.8.1 Informations générales à afficher par le moteur de recherche :

Nombre de documents trouvés :

Cette information permet à l'utilisateur d'évaluer la pertinence de sa requête de recherche. Dans le cas d'un nombre de résultats trop important, il pourra essayer d'affiner sa requête et de lancer ensuite une nouvelle recherche. Dans le cas inverse, il pourra modifier sa requête afin de la rendre plus généraliste et d'obtenir des documents qui lui seront utiles.

Liste des mots ignorés ou dont un équivalent a été trouvé :

Les mots dans cette liste permettront à l'utilisateur de savoir comment sa requête a été analysée et effectuée par le moteur de recherche

II.8.2 Informations spécifiques et détaillées concernant chaque document :

Titre du document indexé

Le titre du document permet à l'utilisateur de se faire une première idée du contenu potentiel du document et donc de son intérêt pour lui

Mesure de la pertinence calculée par le moteur

Cette valeur subjective calculée par le moteur permet d'avoir une évaluation approximative d'un document trouvé par rapport à un autre.

Extrait du document retourné

Il est souhaitable que le moteur de recherche affiche un extrait du document contenant quelques lignes avec le ou les mot-clés recherchés mis en évidence par un sur-lignage par exemple.

URL du lien vers le document d'origine :

Le lien vers le document d'origine sous forme de nommage de type URL, c'est à dire protocole://chemin/nom_document est indispensable afin de l'ouvrir via un navigateur web afin de consulter le document.

Taille du fichier

Cette donnée est optionnelle mais permet de savoir si le document retourné est court comme une plaquette de présentation, un résumé ou si c'est un document plus long comme un livre ou une notice technique.

Date du document

Cette information permet de savoir si le document peut convenir en termes de fraicheur ou s'il est déjà périmé.

II.9 Conclusion

On constate à la lecture des différentes étapes de traitement indiquées précédemment que le moteur de recherche est un logiciel complexe qui nécessite de nombreux traitements des données lors de l'indexation ou de la recherche. Ces traitements nécessitent l'utilisation d'un processus dédié particulier et aussi l'utilisation d'algorithmes spécifiques aux traitements de données textuelles.

III Évaluation des moteurs de recherche et choix d'un des moteurs

Cette partie traitera d'abord de l'évaluation des moteurs de recherche retenus dans le cadre d'une étude et puis ceux-ci seront ensuite analysés au travers d'une méthode d'évaluation de logiciels libres.

III.1 Plan de l'évaluation

Objectifs

Mon travail a consisté à étudier différentes sources Internet (sites web et archive en ligne) pour trouver et retenir les différents moteurs de recherche correspondant au périmètre fonctionnel et technique retenu dans le cahier des charges pour le choix d'un moteur de recherche open source. Comme l'information de ces sources était parfois insuffisante, il m'a fallu analyser la documentation fournie avec les sources des logiciels et parfois même le code source. Avec la totalité de ces renseignements, j'ai pu faire mon étude en me basant sur la méthode d'évaluation QSOS qui sera présentée ci-dessous.

La comparaison entre les différents logiciels a notamment porté sur les aspects suivants :

- Les fonctionnalités d'indexation : les formats de documents indexés (web et bureautiques), l'accès à des données contenues dans un SGBD (Système de gestion de base de données) comme Oracle.
- Les fonctionnalités de recherche : recherche par opérateurs booléens, recherche conceptuelle, recherche approchante, classification automatique
- La gestion de la sécurité et facilité d'intégration : architecture de type « n-tiers », module pour gérer la sécurité ou l'authentification
- La gestion du multilinguisme : jeux de caractères supportés, traitements linguistiques spécifiques.

III.2 Méthode d'évaluation QSOS

La méthode de Qualification et de Sélection de logiciels Open Source (QSOS) est une méthode d'évaluation de logiciels libres. Elle a été développée par la société AtoS à partir de 2004 et placée sous licence libre GFDL (GNU Free Documentation License).

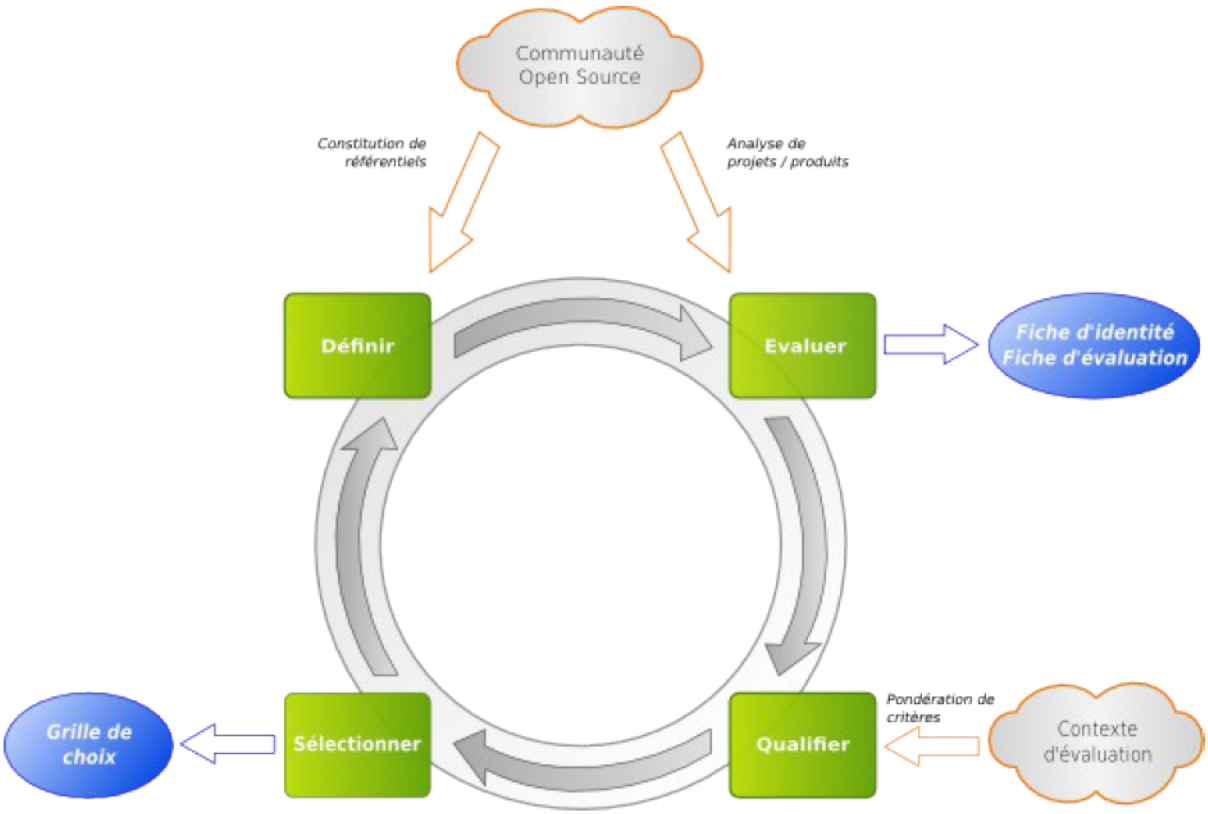

Figure 1 : schéma du processus QSOS
Schéma provenant du site web : http://www.qsos.org/?page_id=3

QSOS consiste en un processus itératif en quatre étapes :

1. Définir les données de référentiel (types de licences, grilles de couverture fonctionnelle par domaine, types de communautés...) ;
2. Évaluer les logiciels selon trois axes principaux : couverture fonctionnelle, risques du point de vue de l'entreprise utilisatrice, risques du point de vue du fournisseur de services (expertise, formation, support). Chaque axe est constitué d'un certain nombre de critères. Par exemple, l'axe des risques pour l'entreprise comprend : la pérennité intrinsèque, l'intégration, l'adaptabilité technique, le niveau

d'industrialisation et la stratégie du projet. Ces critères sont eux-même composés de sous-critères ;

3. Qualifier le contexte spécifique d'une entreprise (ou d'un utilisateur) en effectuant une pondération des critères précédents ;
4. Sélectionner et comparer les logiciels répondant aux besoins.

Ce processus génère des fiches d'identités de logiciels ainsi que des grilles de comparaison et de choix. Les fiches et les grilles fournies par les contributeurs sont également sous licence libre GFDL et disponibles sur le site http://www.qsos.org, afin de permettre leur réutilisation et leur amélioration, gage d'objectivité. La démarche par étape, les multiples critères d'analyse, et la métrologie définis par QSOS en font une méthode d'évaluation objective et argumentée des logiciels libres, précieuse notamment dans les phases amont d'étude d'opportunité de migration vers les logiciels libres, ainsi que pour choisir une solution open source optimale dans un contexte donné.

Dans le cadre de ce mémoire, en l'absence d'une grille d'évaluation de moteur de recherche, j'en ai créé une. J'ai effectué ce travail à partir des critères indiqués dans la documentation de la méthode et en m'inspirant de la grille existante sur le serveur de gestion de base de données Mysql. Cette dernière contient la définition de critères techniques et fonctionnels afin de caractériser ceux qui sont nécessaires à l'évaluation de moteurs de recherche. Cette grille d'évaluation avec un exemple basé sur mnoGoSearch est disponible en Annexe.

Cette méthode permet d'analyser très finement un ensemble de logiciels open source afin de faire un choix objectif et argumenté pour la mise en place de ce type de logiciel au sein d'une entreprise.

Cette méthode nécessite de rechercher beaucoup d'informations et parfois des tests en l'absence d'informations documentées pour remplir correctement les fiches d'évaluation. C'est donc une méthode qui demande du temps à la fois pour créer une grille d'analyse sur un type de logiciel si elle n'existe pas déjà et pour obtenir le nombre important d'informations nécessaires pour garnir une fiche pour un seul logiciel.

III.3 Produits choisis pour l'étude

Après des recherches sur Internet (sources : [WEB-ST], [WEB-MNOGO], [WEB-HTDIG], [WEB-NUTCH], [WEB-BN] et en concordance avec le cahier des charges, les logiciels suivants ont été retenus pour une analyse avec la méthode QSOS :

- Logiciels libres et gratuits à code source ouvert :
  - mnoGoSearch
  - Ht://Dig
  - Nutch
- Logiciel payant et obsolète utilisé à EDF qui est destiné à être remplacé :
  - Verity Search97

Remarque : Cette liste ne contient pas tous les logiciels existants et disponibles sur Internet. Les produits libres présélectionnés ont été sélectionnés surtout sur deux critères principaux :

- logiciel toujours développé et mis à jour régulièrement (sauf pour Ht://Dig dont le développement est arrêté depuis un certain temps mais qui est une référence dans le monde des moteurs de recherche en logiciels libres)
- support du protocole HTTPS.

Le premier critère a été retenu parce qu'en cas de faille de sécurité ou de bogue dans le logiciel, le logiciel doit être corrigé rapidement par les personnes en charge du développement. De plus ce critère permet de choisir les logiciels pérennes et ayant une communauté active pour avoir du support.

Le deuxième critère est une demande exprimée dans le cahier des charges et qui répond à l'augmentation du nombre de sites utilisant le HTTPS à EDF.

III.3.1 mnoGoSearch

mnoGoSearch [WEB-MNOGO] est un moteur d'indexation et de recherche libre sous licence GNU-GPL (GNU General Public Licence) pour les versions Linux et UNIX. Il s'agit d'un produit d'origine russe créé par Alexander Barkov. Le produit était nommé

auparavant « UdmSearch » et a été renommé en mnoGoSearch suite au rachat de ce logiciel par la société Lavtech Corp où travaille actuellement le créateur initial. Il existe des versions payantes pour les systèmes d'exploitation suivants : Windows NT, 2000 et XP. Note : le mot « mnogo » signifie « plusieurs » en russe. Le logiciel a été évalué, testé, mis en place dans le cadre de ce mémoire.

Fonctions d'indexation :

- Capable d'indexer un site web non sécurisé (protocole HTTP 1.0 et 1.1) ou sécurisé (HTTPS), une source de fichiers FTP (protocole FTP), ou un site de Nouvelles (News) (protocole NNTP).
- Capable d'indexer des documents disponibles en local dans un ou plusieurs répertoires sur un système de fichiers géré par le système d'exploitation ou à distance sur d'autres ordinateurs grâce à des protocoles comme NFS.
- Permet l'intégration de connecteurs pour l'indexation de données stockées dans les champs des tables de base de données avec les systèmes de gestion de bases de données (SGBD) suivants : MySQL, PostgreSQL, SQLite, iODBC, Mimer, Virtuoso, Interbase, Oracle , MS SQL, DB2 , Sybase, InterSystems Cache. Il existe aussi des connecteurs pour utiliser des pilotes ODBC variés comme EasySoft ODBC-ODBC bridge ou UnixODBC. La compilation de mnoGoSearch avec UnixODBC en tant que connecteur lui permet d'accéder à une bonne partie des SGBD existants grâce aux pilotes ODBC fournis ou disponibles sur Internet.
- Indexation de pages ou documents web du type HTML, PHP, ASP, JSP qui affichent de l'information encodée en HTML, de type XML ou de documents textuels (plein-texte) c'est à dire sans formatage par des balises (fichiers *.TXT).
- Indexation de l 'information contenue dans des champs texte de fichiers *.mp3.
- Gestion de jeux de caractères sur un ou plusieurs octets : ISO8859-* , UTF-8 (soit 650 langues actuellement), encodage des langues asiatiques (Chine, Taïwan, Japon, Corée), etc..
- Détection de la langue du document indexé, soit 70 langues actuellement
- Possède un segmenteur de mots via des listes pour le chinois, thaï et japonais. Le segmenteur permet d'identifier qu'un ou plusieurs caractères forment un et un seul

mot puisqu'il n'y a pas d'espaces entre les caractères comme dans les langues occidentales.

- Possibilité de stocker une copie compressée du document (hors images) indexé pour obtenir une fonction identique au « En cache » du moteur de recherche Google.
- Support de listes de mots vides fournis en plusieurs langues dont le français afin de réduire la taille des index
- Support des entités HTML
- Détection des documents en doublons
- Support du standard d'exclusion des robots.

Fonctions de recherche :

- Recherche par mots-clés avec des opérateurs booléens. Les opérateurs disponibles sont : ET (&), OU (|) et NEGATION (~) et recherche d'une phrase précise grâce à des mots entre guillemets («).
- Recherche à logique floue : utilisation de fichiers de synonymes, extraction de sous-chaînes (troncature). L'utilisateur peut ajouter sa propre liste de synonymes.
- Recherche par concepts : utilisation de dictionnaires au format ISPELL (anglais, français, italien, russe livrés en standard, nombreuses autres langues disponibles : allemand, polonais, ...) sur Internet. L'utilisateur peut intégrer son propre dictionnaire de lemmatisation ISPELL.
- La pertinence des résultats de recherche peut être affinée en modifiant la pondération de plusieurs paramètres (date, nombre d'occurrences du ou des mots recherchés, proximité des mots, taille du document, etc..)
- Recherche sur des balises HTML spécifiques : TITLE, BODY, DESCRIPTION, et KEYWORD
- Recherche insensible aux accents possible
- Pertinence très fortement paramétrable dans le formulaire de recherche
- Recherche par catégories

Fonctions pour l'intégration et la sécurité :

- Architecture n-tiers : un « template » HTML personnalisable est fourni pour la recherche et la présentation de la liste des résultats. Ce « template » a été modifié par l'auteur de ce document afin d'ajouter l'internationalisation du formulaire, c'est à dire d'avoir le formulaire en français grâce au chargement d'un fichier avec les traductions et est désormais disponible en français, anglais et polonais.
- Le logiciel d'indexation est écrit en C et le module de recherche est disponible en C (CGI), PHP et PERL pour intégration dans différents types de site web.
- Sécurité : elle s'appuie sur la sécurité du serveur web, c'est à dire sur les différentes authentifications possibles via un compte utilisateur et un mot de passe qui sont définis au niveau du serveur Web (Apache ou IIS) afin de pouvoir indexer la ou les zones protégées. Il est donc possible d'utiliser une authentification à l'aide d'un fichier contenant un identifiant et un mot de passe ou utilisant LDAP (Lightweight Directory Access Protocol) grâce au module mod_ldap pour Apache. mnoGoSearch fonctionne aussi très bien avec des sites web sécurisés utilisant des « cookies » pour les sessions.
- Support des « proxies » avec authentification pour indexer des sites sur Internet.
- Supporte une architecture de type « cluster ». Il est possible, dans le contexte d'une architecture distribuée, de répartir l'index des sites web dans plusieurs bases de données stockées sur plusieurs machines.
- Mise en surbrillance des termes de la recherche dans le document résultat

Sites exemples :

- Mysql (www.mysql.com)
- Eclipse (www.eclipse.org)
- Ministère de l'agriculture et de la pêche (www.agriculture.gouv.fr)

Fonctions absentes ou incomplètes :

- Peu de formats de documents pris en compte de façon native mais ces derniers sont des outils à jour et performants.
- Indexation de documents bureautiques avec des parseurs externes performants : Acrobat (*.PDF) avec Xpdf, Rich Text Format (*.RTF) avec UnRTF, MS Word (*.DOC) , MS Excel (*.XLS) , MS Powerpoint (*.ppt) avec Catdoc pour ces 3 derniers formats, Postscript (*.ps) avec Ghostscript, ODF (Open Document Format) (*.ODT) avec ODFTools, Microsoft Word 2007 et supérieur (*.docx) avec doc2txt, Microsoft Powerpoint 2007 et supérieur (*.pptx) avec pptx2txt, Microsoft Excel 2007 et supérieur (*.xlsx) avec xlsx2csv, WordPerfect (*.WPD) avec libpwd, MS Works avec libwps.
- Pas de support de la recherche avec sensibilité à la casse.
- Pas de support de la recherche avec des expressions régulières
- L'administration se fait en ligne de commande
- Pas de réelle automatisation de la mise à jour de l'index au gré des mises à jour / suppressions de documents

Conclusion :

Adaptation au cahier des charges : Moyenne à forte

III.3.2 Nutch

Nutch [WEB-NUTCH] est un moteur de recherche libre sous licence GNU-GPL (GNU General Public Licence) développé en Java. Il s'agit d'un logiciel créé par Doug Cutting qui est l'initiateur et le coordinateur de ce projet. Il utilise Lucene comme bibliothèque de moteur de recherche et d'indexation. En revanche, le robot de collecte a été créé spécifiquement pour ce projet.

Fonctions d'indexation :

- Capable d'indexer un site web non sécurisé (protocole HTTP 1.0 et 1.1) ou sécurisé (HTTPS), une source de fichiers FTP (protocole FTP)

- Capable d'indexer des documents disponibles en local dans un ou plusieurs répertoires sur un système de fichiers géré par le système d'exploitation.
- Indexation de pages ou documents web du type HTML, PHP, ASP, JSP qui affichent de l'information encodée en HTML, de type XML, des documents textuels (plain-texte) c'est à dire sans formatage par des balises (fichiers *.TXT) ou d'autres types (Java, WAP, RSS,SWF,NEWS,CSS,SGML).
- Indexation de documents bureautiques avec des parseurs internes : Acrobat (*.PDF), Rich Text Format (*.RTF), MS Word (*.DOC), MS Excel (*.XLS), MS Powerpoint (*.ppt), Open Document Format (*.ODF), StarOffice (*.SXW), PS .
- Support à l'indexation de documents compressés avec les formats suivants : Zip, Gzip, Bzip2.
- Gestion de jeux de caractères sur un ou plusieurs octets : ISO8859-* , UTF-8 (soit 650 langues actuellement), encodage des langues asiatiques (Chine, Taïwan, Japon, Corée), etc..
- Détection de la langue du document indexé, soit 14 langues d'Europe de l'Ouest uniquement.
- Support de listes de mots vides afin de réduire la taille des index, mais aucune liste n'est fournie.
- Support des entités HTML
- Détection des documents en doublons
- Support du standard d'exclusion des robots.

Fonctions de recherche :

- Recherche par mots-clés avec des opérateurs booléens. Les opérateurs disponibles sont : OU (+) et NEGATION (~) et recherche de phrase précise grâce à des mots entre guillemets («).
- La pertinence des résultats de recherche peut être affinée en modifiant la pondération de plusieurs paramètres (date, nombre d'occurrences du ou des mots recherchés, proximité des mots, taille du document, etc..)
- Support de la recherche avec sensibilité à la casse.

- Pertinence paramétrable dans le fichier de configuration de la recherche

Fonctions pour l'intégration et la sécurité :

- Architecture n-tiers, un « template » HTML personnalisable est fournie pour la recherche et la présentation de la liste des résultats. Le modèle (« template ») est internationalisé en 18 langues.
- Le logiciel d'indexation est écrit en Java et le module de recherche est en Java et Jsp pour intégration dans différents types de site web.
- La sécurité : s'appuie sur la sécurité du serveur web, c'est-à-dire sur les différentes authentifications possibles via un compte utilisateur et un mot de passe qui sont définis au niveau du serveur Web (Apache ou IIS) afin de pouvoir indexer la ou les zones protégées.
- Support des proxies avec authentification pour indexer des sites sur Internet.
- Supporte une architecture de type « cluster ». Il est possible, dans le contexte d'une architecture distribuée, de répartir l'index sur plusieurs machines.

Sites exemples :

- Mister Bot (http://www.misterbot.fr/)
- Oregon State University (http://search.oregonstate.edu/)

Fonctions absentes ou peu développées :

- Pas de segmenteur pour les langues asiatiques
- Pas de recherche à logique floue
- Pas de recherche par concepts
- Pas de recherche sur des balises HTML spécifiques
- Pas de recherche insensible aux accents
- Pas de support de la recherche avec des expressions régulières
- L'administration se fait en ligne de commande

- Pas d'indexation de bases de données
- Pas de réelle automatisation de la mise à jour de l'index au gré des mises à jour / suppressions de documents
- Mise en place du logiciel très complexe (difficulté de mise en place du pseudo système de fichiers HDFS)
- Industrialisation du logiciel difficile (possibilité de créer une collection avec un seul fichier de configuration pour un seul site web)

Conclusion :

Adaptation au cahier des charges : Faible à moyenne

III.3.3 HtDig

Ht://Dig [WEB-HTDIG] est un moteur de recherche et d'indexation libre développé par l'université de San Diego sous licence GNU-GPL (General Public Licence). Il fonctionne sur plusieurs systèmes d'exploitation : Solaris, HP/UX, IRIS, SunOS, Linux, MacOS X (pas de version Windows). Le code source C++ est disponible sur le site Internet. On le retrouve dans des distributions de Linux. Ce logiciel n'est plus développé ni mis à jour, les développeurs de la version 4.0 de HtDig 4.0 ne l'ont toujours pas sortie.

Points forts / capacités d'indexation :

- Capable d'indexer un site web non sécurisé (protocole HTTP) ou sécurisé (HTTPS)
- Capable d'indexer des documents disponibles en local dans un ou plusieurs répertoires sur un système de fichiers géré par le système d'exploitation
- Indexation de pages ou documents web du type HTML, PHP, ASP, JSP qui affichent de l'information encodée en HTML ou des documents textuels (plein-texte) c'est à dire sans formatage par des balises (fichiers *.TXT)
- Support des entités HTML

- Points forts : capacités de recherche
- Recherche par mots-clés avec des opérateurs booléens. Les opérateurs disponibles sont : ET (AND, OU (OR) et recherche de phrase précise grâce à des mots entre guillemets («)
- Recherche à logique floue : utilisation de fichiers de synonymes, extraction de sous-chaînes (troncature). L'utilisateur peut ajouter sa propre liste de synonymes. Tolérance sur orthographe approximative selon plusieurs algorithmes (opérateurs « soundex » et « métaphone »)
- Recherche par concepts : utilisation de dictionnaires au format ISPELL L'utilisateur peut intégrer son propre dictionnaire de lemmatisation ISPELL
- Recherche sur des balises HTML spécifiques : TITLE, BODY, DESCRIPTION, et KEYWORD
- Recherche insensible aux accents possible
- Pertinence paramétrable dans le formulaire de recherche
- Points forts : aspects intégration et sécurité
- Architecture n-tiers, un modèle (« template ») HTML personnalisable est fourni pour la recherche et la présentation de la liste des résultats
- La sécurité : s'appuie sur la sécurité du serveur web, c'est à dire sur les différentes authentifications possibles via un compte utilisateur et un mot de passe qui sont définis au niveau du serveur Web (Apache ou IIS) afin de pouvoir indexer la ou les zones protégées
- Sait s'intégrer dans une architecture de type cluster. Il est possible, dans le contexte d'une architecture distribuée, de répartir l'index d'un fonds documentaire sur plusieurs machines.

Sites exemples :

- http://www.pyrenet.fr/htdig/search.html (3.1.5)
- http://listes.linuxarverne.org/htdig/search.html (3.1.6)
- http://didier.quartier-rural.org/elucu/htdig-vf/lisezmoi.html
- http://search-fr.cirad.fr/htdig/search.html (3.2.0 beta6)

Points faibles déduits de l'étude :

- Peu de formats de documents pris en compte de façon native mais ces derniers sont des outils à jour et performants
- Indexation de documents bureautiques avec des parseurs externes performants : Acrobat (*.PDF) avec Xpdf, Rich Text Format (*.RTF) avec UnRTF, MS Word (*.DOC) , MS Excel (*.XLS) , MS Powerpoint (*.ppt) avec Catdoc pour ces 3 derniers formats, Open Document Format (*.ODF) avec ODFReader
- Pas d'indexation de bases de données
- Pas de recherche par catégories
- Pas de support de l'UTF-8 et de l'encodage des langues asiatiques
- Pas de détection de la langue
- Pas de segmenteur pour les langues asiatiques
- Gestion de la sécurité limitée
- Pas de réelle automatisation de la mise à jour de l'index au gré des mises à jour / suppressions de documents.

Conclusion :

Adaptation au cahier des charges : Faible à moyenne.

De plus on note les problèmes suivants :

- Plus de développement du logiciel
- Beaucoup de sites web Internet possédant des pages prévues pour la recherche HtDig qui ne fonctionnent plus.

III.3.4 Verity Search'97

Descriptif :

VERITY INC [WEB-VERITY] était l'un des acteurs principaux dans le secteur des moteurs de recherche avant d'être racheté par la société Autonomy en Décembre 2005.

Cette dernière était une entreprise américaine fondée en 1988. Cette société dotée d'une filiale française recensait plus de mille clients dans le monde, dont des sites d'E-commerce, de places de marchés, des administrations gouvernementales, et des fournisseurs de service en ligne. En France, Verity était présente chez les grands comptes suivants AFP, Airbus, Dassault, EDF-GDF, DGA, Thomson...

VERITY Search'97 était le moteur d'indexation et de recherche (payant) de la société VERITY INC. Le logiciel est compatible avec les plateformes Windows NT4, Unix: Sun Solaris 2.5, 2.6, 2.8, DEC Alpha UNIX 4.0, Hewlett-Packard HP/UX 10, SGI IRIX 6 , IBM RS/6000 AIX 4.

L'étude des trois logiciels libres précédents se fait par rapport à ce logiciel qui doit être remplacé par un produit plus à jour que ce produit datant de 1998. Le but est notamment d'éviter toute régression pour les utilisateurs.

Fonctions d'indexation :

- Capable d'indexer un site web non sécurisé (protocole HTTP) ou sécurisé (HTTPS)
- Le support du HTTPS était une fonction payante
- Capable d'indexer des documents disponibles en local dans un ou plusieurs répertoires sur un système de fichiers géré par le système d'exploitation
- Permet l'intégration de connecteurs pour l'indexation de données stockées dans des bases Lotus Notes. Le support de cette fonction était payant
- Indexation de pages ou documents web du type HTML, PHP, ASP, JSP qui affichent de l'information encodée en HTML et des documents textuels (plain-texte) c'est à dire sans formatage par des balises (fichiers *.TXT)
- Indexation de documents bureautiques avec des parseurs internes obsolètes : Acrobat v3 (*.PDF), Rich Text Format (*.RTF), MS Word (*.DOC) , MS Excel (*.XLS) , MS Powerpoint (*.ppt), WordPerfect (*.wpd), Lotus 1-2-3, MS-Works .
- Gestion de jeux de caractères sur un ou plusieurs octets : ISO8859-1 à 9, Windows1251 à 1258, UTF-8 (soit 650 langues actuellement), encodage des langues asiatiques (Chine, Taïwan, Japon, Corée), etc...

- Détection de la langue du document indexé mais nécessite la cartouche de langue appropriée. Chaque cartouche pour une langue était payante
- Support de listes de mots vides fournis en plusieurs langues afin de réduire la taille des index. Chaque cartouche pour une langue était payante
- Support des entités HTML
- Détection des documents en doublons
- Support du standard d'exclusion des robots.

Fonctions de recherche :

- Recherche par mots-clés avec des opérateurs booléens. Les opérateurs disponibles sont : ET (AND), OU (OR), NEGATION (NOT), proche <NEAR>, phrase <PHRASE> et recherche de phrases précises grâce à des mots entre guillemets («)
- Recherche à logique floue : support de Soundex, Thesaurus (recherche de synonymes), fonctions TYPO et WORD sur une orthographe approximative
- Recherche par concepts : support de stemming (lemmatisation) en utilisant les doubles quotes, on recherche sur le mot et ses racines
- La pertinence des résultats de recherche peut être affinée en modifiant le poids des mots dans la requête de recherche
- Recherche sur des balises HTML spécifiques : TITLE, BODY, DESCRIPTION, et KEYWORD
- Support de la recherche avec sensibilité à la casse
- Utilisation de jokers pour la recherche (comme gaz* ou ga?).

Fonctions pour l'intégration et la sécurité :

- Architecture n-tiers, un modèle (« template ») HTML personnalisable est fourni pour la recherche et la présentation de la liste des résultats. Ce modèle est disponible en français et en anglais.
- Le logiciel d'indexation est en C et le module de recherche est disponible en C (CGI) pour intégration dans différents types de site web.

- Sécurité : elle s'appuie sur la sécurité du serveur web, c'est à dire sur les différentes authentifications possibles via un compte utilisateur et un mot de passe qui sont définis au niveau du serveur Web (Apache ou IIS) afin de pouvoir indexer la ou les zones protégées.
- Support des proxies avec authentification pour indexer des sites sur Internet.
- Mise en surbrillance des termes de la recherche dans le document résultat.

Sites exemples :

- Norm d'EDF (http://norm.edf.fr)

Fonctions absentes ou peu complètes:

- Pas d'indexation de fichiers XML.
- Pas de support du caractère Euro dans les jeux d'encodage
- Pas de segmenteur en standard pour les langues asiatiques
- Pas de possibilité de stocker une copie compressée du document (hors images) indexé pour obtenir une fonction identique au « En cache » du moteur de recherche Google.
- Pas de support du format Open Document Format (*.ODF)
- Pas de support de la recherche insensible aux accents
- Pas de support des cookies
- Pas de support de la recherche avec des expressions régulières
- L'administration se fait en ligne de commande
- Pas de réelle automatisation de la mise à jour de l'index au gré des mises à jour / suppressions de documents

III.3.5 Récapitulatif de l'étude

Dans cette partie, l'objectif principal sera de présenter de manière synthétique via un tableau le résultat de l'étude de ce chapitre.

La méthode QSOS présentée au début de ce chapitre m'a permis de faire une évaluation objective des différents moteurs parce qu'elle indique déjà au préalable un ensemble d'informations à trouver et à évaluer selon trois axes principaux pour chaque logiciel : couverture fonctionnelle, risques du point de vue de l'entreprise utilisatrice, risques du point de vue du fournisseur de services (expertise, formation, support). Cette méthode fournit, avec son modèle de grille d'évaluation, un ensemble de critères importants à évaluer et permet donc d'éviter d'en oublier. Ensuite en fonction du type du logiciel à évaluer, il faut réutiliser ou améliorer une grille existante, mais dans mon cas il n'y avait rien et j'ai donc dû déterminer, par différentes recherches sur Internet, un ensemble de critères les plus courants et assurant le maximum de couverture des fonctionnalités. Mon retour d'expérience me prouve que cette méthode a donc le mérite d'être objective et fiable mais demande beaucoup de temps pour trouver les informations nécessaires. Au final, on obtient une comparaison détaillée des logiciels du domaine choisi qui permet de faire facilement un choix en fonction des besoins réels des utilisateurs.

Dans la partie précédente, j'ai détaillé chaque logiciel retenu en donnant pour chacun un descriptif, les fonctions d'indexation, les fonctions de recherche, les fonctions pour l'intégration et la sécurité, et les fonctions absentes ou complètes. L'ensemble des informations ont ensuite été renseignées dans les grilles QSOS pour faire une synthèse.

Le tableau ci-dessous synthétise les principaux résultats de cette étude.

LOGICIELS	Mnogosearch	Nutch	HtDig	Search'97
CRITERES DE L'ETUDE	1	2	3	4
FONCTIONS D'INDEXATION				
Indexation de documents en code HTML	X	X	X	X
Indexation de documents Ms Office (xls, doc, ppt)	X	X	X	X
Indexation de documents Ms Office 2007 (xlsx, docx, pptx)	X	X	X	
Indexation de documents Adobe Acrobat (pdf)	X	X	X	X
Indexation d'un site web en HTTP	X	X	X	X
Indexation d'un site web en HTTPS	X	X	X	Option
Indexation de bases de données	X			
Indexation d'un système de fichiers	X	X	X	X
Listes de mots vides en plusieurs langues	X			X
Segmenteur pour langues asiatiques	X			
CAPACITE DE RECHERCHE				
Recherche booléenne (par mots-clés)	X	X	X	X
Recherche à logique floue	X		X	X
Recherche par concepts	X		X	X
Recherche par navigation (catégories)	X			
INTEGRATION ET SECURITE				
Architecture n-tiers	X	X	X	X
Templates HTML fournis	X	X	X	X
Intégration possible en C,Java, PHP, ou PERL	X	X	X	X
Sécurité via un identifiant/mot de passe UNIX, LDAP	X	X-	X-	X-
Support des cookies	X	X	X	
Mise en surbrillance des résultats de recherche	X		X	

Tableau 1 : Récapitulatif de l'étude

Légende :

X : Fonction présente ou l'indexer peut utiliser un parseur externe tiers existant par configuration

X+ : Fonction présente et développée ou exploitable de suite

X- : Fonction présente mais peu développée ou nécessitant un développement

Le tableau ci-dessus indique dans sa partie gauche l'ensemble des principales fonctions de chacun des moteurs de recherche évalués. Ces derniers sont précisés dans la première ligne du tableau. Les X dans les cases indiquent si la fonction est présente dans le logiciel et à quel état de maturité suivant la légende sous ce tableau.

III.3.6 Conclusion et choix du moteur

Le produit retenu est mnoGoSearch. Il offre des mises à jour régulières au contraire de HtDig. De plus ce logiciel est facilement industrialisable à l'inverse de Nutch. Ce critère est important dans le contexte d'une grande entreprise comme EDF où il y a beaucoup de sites intranet. mnoGoSearch a un excellent support des langues étrangères et, de plus, est facilement extensible sur ce point. Il gère aussi très bien l'aspect sécurité côté serveur web grâce au protocole HTTPS, à l'authentification basique et à des cookies. La méthode QSOS a rempli son rôle en permettant de sélectionner le moteur retenu grâce à une liste de critères objectifs définie de manière objectif. Malgré le temps nécessaire pour trouver les critères d'évaluation, rédiger les grilles avec les bonnes informations qui sont sourcées pour vérifier les informations, puis pondérer l'évaluation en fonction des critères les plus importants en fonction de l'usage prévu du moteur à choisir, j'obtiens une excellente évaluation qu'il serait encore possible d'améliorer en ajoutant des critères de fonctions plus spécialisées.

IV Présentation fonctionnelle du moteur de recherche choisi

IV.1 Présentation du mode de recherche

Le moteur de recherche mnoGoSearch est fourni avec un modèle de recherche personnalisable qui permet de répondre aux besoins des 2 publics les plus courants. D'un côté, les utilisateurs qui sont à la recherche d'une interface graphique simple et épurée avec un seul champ de recherche et un bouton pour envoyer la recherche dans le formulaire, c'est-à-dire « à la Google » avec sa page d'accueil standard. Ils auront donc une page web pour une recherche dite "simple". De l'autre, des utilisateurs qui maîtrisent plus les différentes fonctionnalités des moteurs de recherche et auront une interface graphique plus évoluée avec les même éléments que la recherche simple, c'est-à-dire avec en plus plusieurs listes déroulantes afin d'affiner la recherche suivant différents critères dans le formulaire. Ceux-ci auront donc une page web pour une recherche dite "avancée".

IV.1.1 Recherche simple

La recherche simple peut se représenter sous la forme du diagramme UML (Unified Modeling Language) suivant :

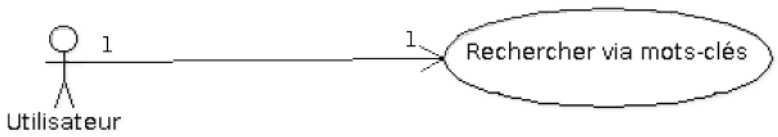

Figure 2 : Diagramme UML de cas d'utilisation de la recherche simple

Sur ce diagramme UML de cas d'utilisation standard, nous avons les opérations décrites ci-dessous.

Un utilisateur peut lancer une ou plusieurs recherches sur des mots-clés dans le moteur de recherche afin d'obtenir des résultats et d'accéder à des documents correspondant à l'information souhaitée.

Plus concrètement cette recherche simple correspond au formulaire ci-dessous.

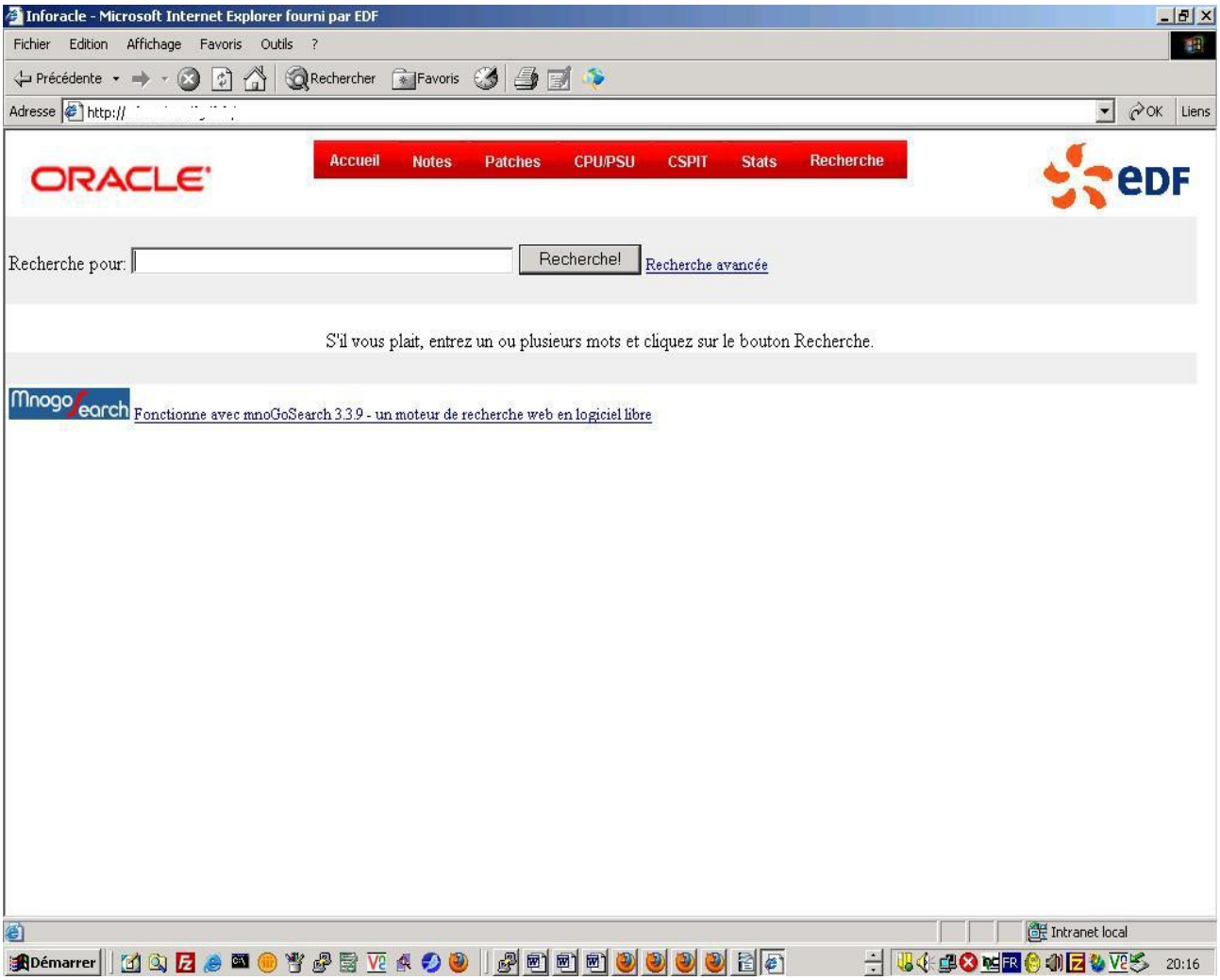

Figure 3 : capture d'écran d'un formulaire de recherche simple

Sur le formulaire ci-dessus, on voit que le haut de l'écran est habillé par un bandeau avec une charte graphique qui comprend des logos et le menu du site web.

Sous cet ensemble, nous trouvons le formulaire de recherche au sens strict du terme avec un champ de recherche et à sa droite un bouton qui permet à l'utilisateur d'envoyer la requête de recherche au serveur. Sous ces éléments de la page web, nous avons des

informations pour l'utilisateur, et en pied de page, le logo et une information sur le moteur de recherche utilisé.

Il est à noter que l'utilisateur peut entrer différents mot-clés dans le champ de recherche mais le moteur de recherche ne permet pas de faire de recherche sans mot-clé. Et donc dans ce cas, il n'est pas possible de connaître le nombre de documents indexés par le moteur de recherche, ce qui peut aussi être une sécurité.

IV.1.2 Recherche avancée

La recherche avancée peut se représenter sous la forme du diagramme UML suivant :

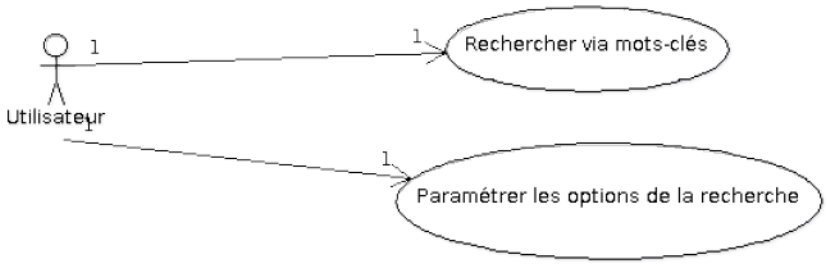

Figure 4 : Diagramme UML de cas d'utilisation de la recherche avancée

Sur ce diagramme de cas d'utilisation standard, comme dans le cas de la recherche simple, un utilisateur peut lancer une ou plusieurs recherches sur des mot-clés dans le moteur de recherche afin d'obtenir des résultats et d'accéder à des documents correspondant à l'information souhaitée.

Dans le cas de la recherche avancée, l'utilisateur a en plus la possibilité de paramétrer la recherche qu'il va faire via différentes options.

Plus concrètement cette recherche avancée correspond au formulaire ci-dessous.

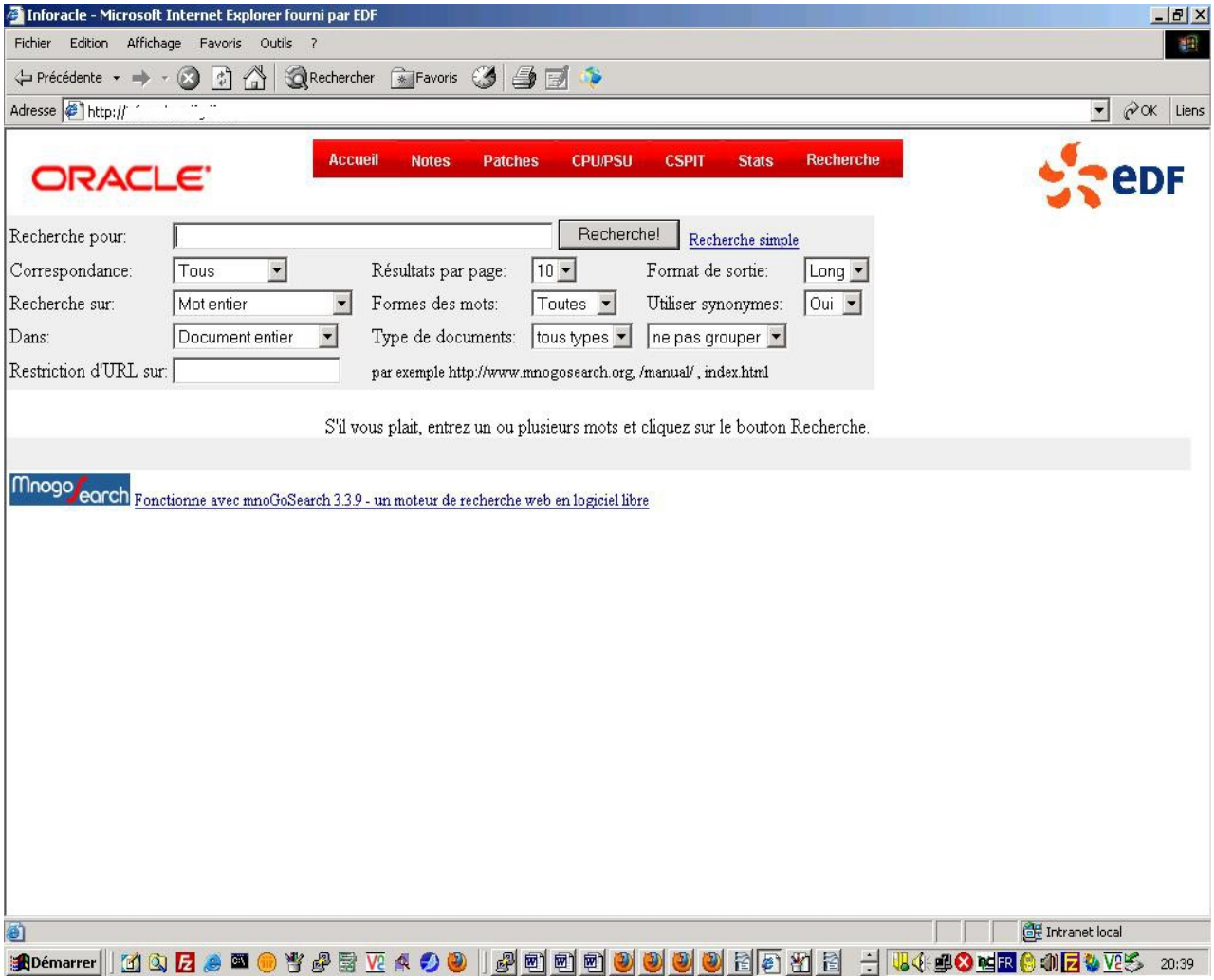

Figure 5 : capture d'écran d'un formulaire de recherche avancée

Comme dans le cas de la recherche simple, nous pouvons voir que, sur le formulaire ci-dessus, le haut de l'écran est habillé par un bandeau avec une charte graphique qui comprend des logos et le menu du site web.

Sous cet ensemble, nous trouvons le formulaire de recherche au sens strict du terme avec un champ de recherche et, à sa droite, un bouton qui permet à l'utilisateur d'envoyer la requête de recherche au serveur. Sous ces éléments de la page web, nous avons des informations pour l'utilisateur et, en pied de page, le logo et une information sur le moteur de recherche utilisé.

Il est à noter que l'utilisateur peut fournir différents mots-clés dans le champ de recherche mais le moteur de recherche ne permet pas de faire de recherche sans mot-clé. Et

donc dans ce cas, il n'est pas possible non plus de connaître le nombre de documents indexés par le moteur de recherche, ce qui peut aussi être une sécurité.

La recherche avancée offre différentes options supplémentaires que nous allons décrire :

- « Recherche sur » : cette option offre les possibilités Mot entier (recherche sur le mot en entier), Mot débutant (recherche sur tous les mots commençant par le mot-clé), Mot finissant (recherche sur tous les mots finissant par le mot-clé), Sous chaîne d'un mot (recherche sur tous les mots dont une partie contient le mot-clé)
- « Correspondance »: cette option offre les possibilités « Tous » ou « Quelconque », elle permet de choisir entre des résultats sur le ou les mot-clés entrés ou sur des mots approchant ce ou ces mot-clés.
- « Dans » : cette option offre les possibilités « Document entier » (recherche dans le document en entier du mot-clé indiqué), « Description » (recherche dans la description, c'est à dire les propriétés du document du mot-clé indiqué), « Mots-clés » (recherche dans le champ prévu pour les mots-clés du document du mot-clé indiqué), « Titre » (recherche dans le titre document du mot-clé indiqué) , « Corps » du document (recherche dans le corps du document, c'est à dire hors titre, description, champ mots-clés du document du mot-clé indiqué)
- « Restriction d'URL » sur : ce champ permet d'indiquer sur quelle URL (d'un site web ou autres sources de documents) la recherche de documents doit être restreinte.
- « Résultats par page » : cette option offre par défaut la possibilité d'afficher 10, 20 ou 50 résultats de recherche par page web de résultat.
- « Formes des mots » : cette option offre les possibilités « Toutes » ou « Exactes », elle permet de choisir entre avoir des résultats sur le ou les mot-clés entrés ou sur des mots approchant ce ou ces mot-clés rentrés.
- « Types de documents » : cette option permet de restreindre la recherche sur un type MIME (identifiant de format de données sur Internet) déterminé de documents, ici les choix proposés sont « tous types » (recherche sur la totalité des documents), « text/html » (recherche uniquement sur les documents au format HTML), « text/plain » (recherche uniquement sur les documents au format texte brut sans aucun formatage, habituellement fichiers avec l'extension *.txt).

Cette option de recherche peut être améliorée, il faut alors ajouter dans le modèle une recherche sur d'autres types MIME communs comme : pdf, doc, ppt, xls, odt.

- En face de cette option, il y en a une autre sans titre qui possède 2 choix possibles « ne pas grouper » (les documents provenant de différents sites web ou sources URL sont affichés par ordre de pertinence) et « grouper par site » (les documents sont affichés par groupe avec uniquement le nom de leur source ou du site web)
- « Format de sortie » : cette option permet d'avoir une information plus ou moins détaillée pour chaque résultat de recherche. « Long » (affiche le titre, l'URL et un extrait du texte du document contenant le ou les mots-clés) , « Court » (affiche uniquement le titre du document pour chaque résultat de recherche), « URL » (affiche uniquement l'URL du document pour chaque résultat de recherche).
- « Utiliser synonymes » : cette option permet d'utiliser le dictionnaire des synonymes afin de faire une recherche sur le mot-clé et ses synonymes. Exemple : utiliser « voiture » comme mot-clé et faire une recherche aussi sur les documents contenant le mot-clé « automobile ».

  L'utilisation de cette option rend plus long l'affichage des résultats de la recherche.

Il existe une autre option de recherche non présentée ici puisque non utilisée qui permet de faire une recherche spéciale en restreignant la recherche sur un ou plusieurs sites en particulier en créant une ou plusieurs « collections » de sites web.

IV.1.3 Taches de l'administrateur du moteur de recherche

L'administrateur peut déclencher l'indexation des sites web ou d'autres sources de documents. Il a aussi en charge la configuration initiale puis la mise à jour dans le temps du paramétrage de l'indexation d'une ou plusieurs sources de documents. En plus de l'indexation, l'administrateur doit effectuer les tâches citées précédemment sur le ou les formulaires de recherche et les paramètres de recherche.

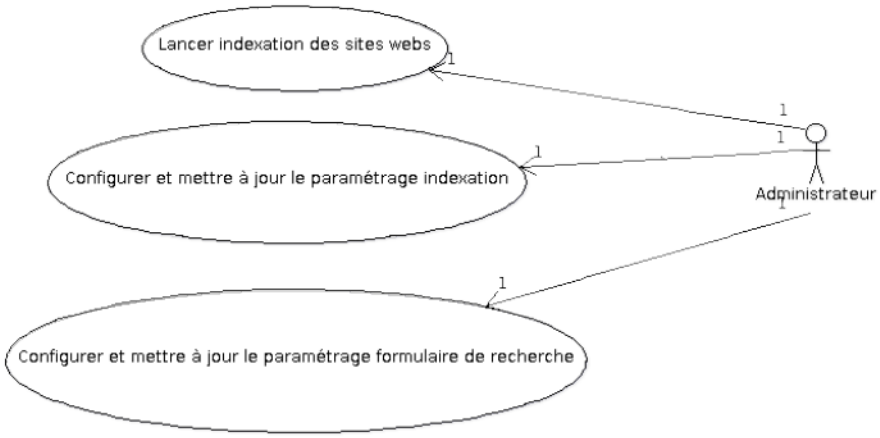

Figure 6 : Diagramme UML de cas d'utilisation des tâches de l'administrateur

IV.1.4 Résultats de recherche

L'utilisateur, après avoir lancé une recherche avec un ou plusieurs mots-clés et différentes options de recherche dans le cas de la recherche avancée, obtient l'écran ci-dessous avec le mot-clé Oracle.

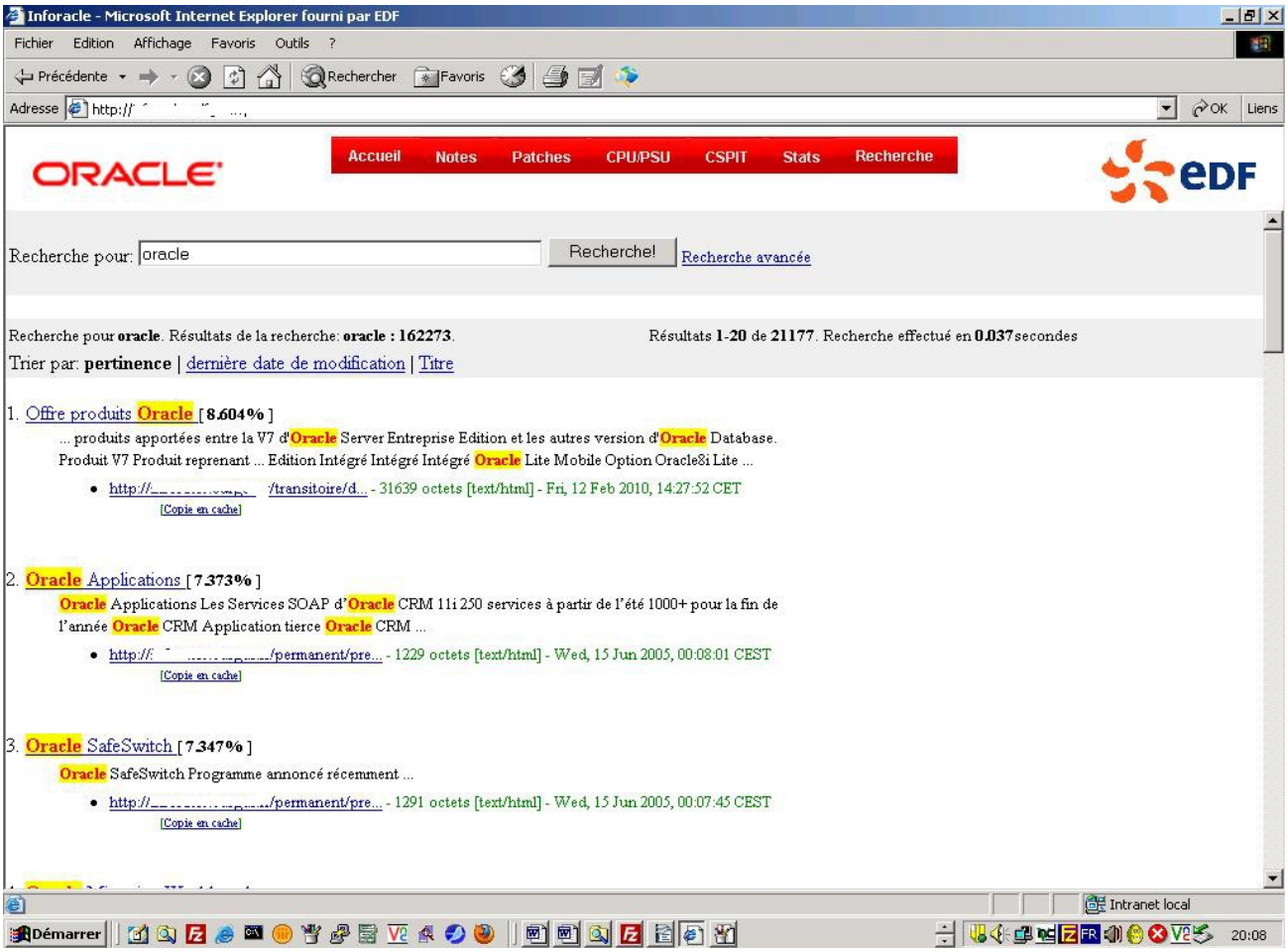

Figure 7 : capture d'écran des résultats de la recherche avec mot-clés en surbrillance

Nous avons différentes informations sur la page de résultats que nous allons détailler de haut en bas.

La première ligne sous le champ de recherche indique les mots-clés qui ont servi à la recherche et le nombre de résultats (occurrences totales) trouvés pour chacun de ces mots-clés. En fin de ligne, le nombre de documents trouvés et le temps mis pour faire la recherche sont affichés.

Ensuite la page d'affichage des résultats permet de trier les résultats de la recherche suivant trois critères :

- « pertinence » : cette pertinence est calculée par le moteur de recherche et est affichée sous forme de pourcentage en face de chaque résultat. Elle est calculée en fonction de différents critères comme le nombre d'occurrences dans le document, la présence ou non de mots-clés dans le titre ou le champ mots-clés de ce document.

- « Dernière date de modification » : les documents les plus récents sont affichés au début de la liste des résultats. Il est à noter que cette date n'est pas correcte pour les documents qui sont générés dynamiquement comme les pages web en PHP ou JSP.
- « Titre » : les documents ayant le ou les mots-clés recherchés dans leur titre sont affichés en premier.

Ensuite nous avons la liste des résultats classés ici par pertinence. Pour ce type de tri, nous avons chaque résultat qui est affiché de la manière suivante :

- Un numéro dans l'ordre des résultats suivi du titre du document et du pourcentage de pertinence calculé par le moteur de recherche.
- Ensuite sous cette ligne, le moteur de recherche a extrait une ou plusieurs lignes du document contenant un ou plusieurs mots-clés recherchés qui sont mis en évidence via une surbrillance. On peut considérer cela comme une sorte de « résumé » du document.
- Sous ce « résumé », l'URL du document est affichée suivie par d'autres informations sur ce même document : taille en octets, type mime, jour dans la semaine, date et heure. Tous ces éléments facilitent la contextualisation du document pour l'utilisateur.
- Le lien « Copie en cache » à la fin du document permet d'afficher une copie du document mise en forme comme ci-dessous.

Si la lemmatisation via Ispell était activée avec le support des synonymes, dans ce cas dans la ligne « Résultats de recherche » ci-dessus, nous aurions par exemple pour le mot-clé voiture, l'affichage du nombre d'occurrences trouvées du mot voiture s'il y en avait, suivi du nombre d'occurrences trouvées du mot automobile s'il y en avait, puisque le mot automobile est un synonyme de voiture.

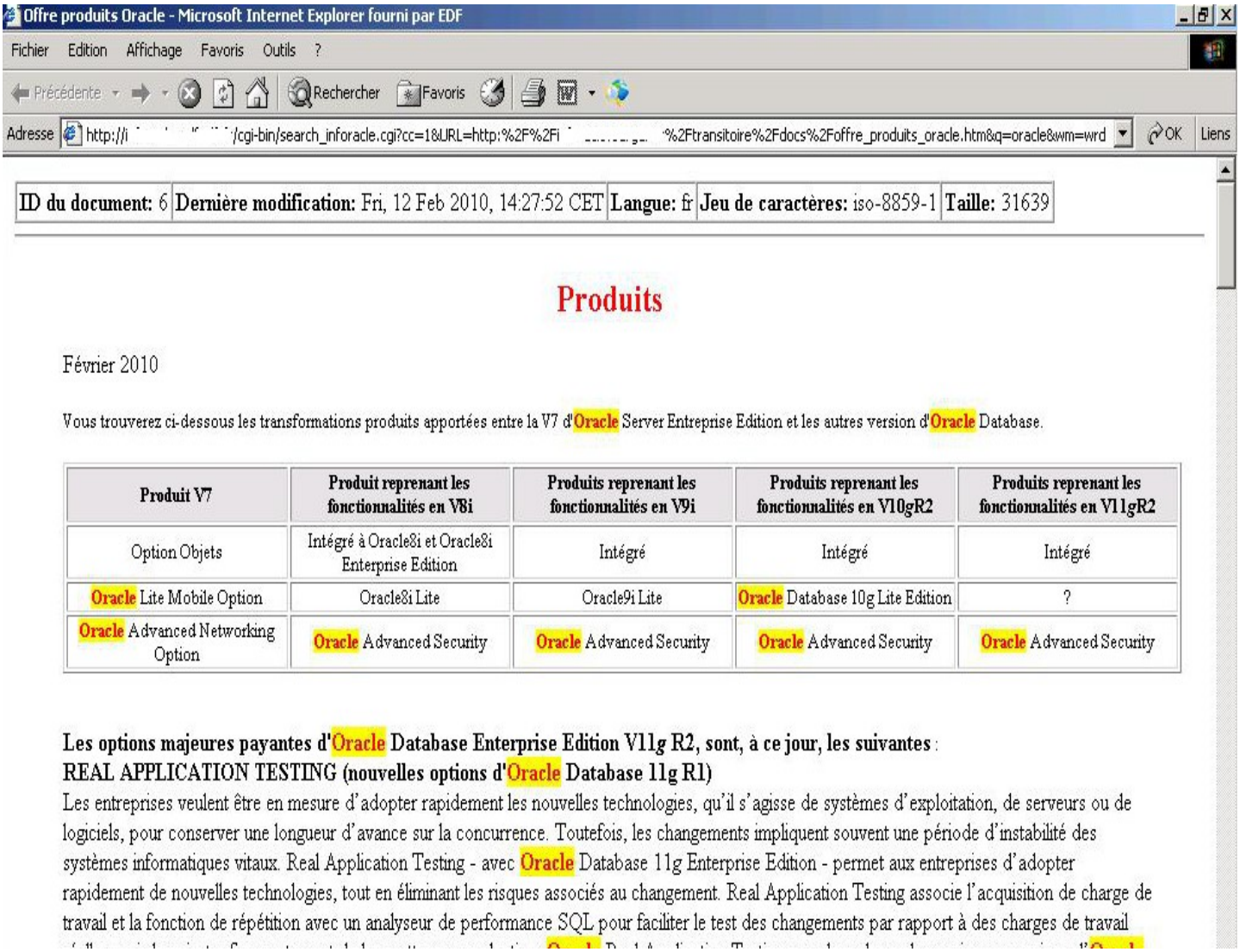

Figure 8 : capture d'écran d'une page en cache avec mot-clés en sur-brillance

Sur le document ci-dessus, qui est une copie en cache du document, on voit que le document est remis en forme avec un chapeau qui contient les informations suivantes : « ID du document » avec son numéro pour le moteur de recherche, (Date de) « Dernière modification » suivie de la date au format anglais (Nom du jour, Jour Mois Année), « La langue » (du document) suivie de son code sur 2 lettres, le « Jeu de caractères » utilisé suivi de son numéro, « Taille du document » suivie du chiffre de sa taille en octets.

Sous ces éléments, nous retrouvons le texte d'origine avec sa mise en page de manière globale mais sans les images et avec les mots-clés en surbrillance afin de faciliter la lecture.

La navigation dans la page des résultats se fait via des liens représentés par des numéros en bas de la page des résultats comme montré sur l'image ci-dessous.

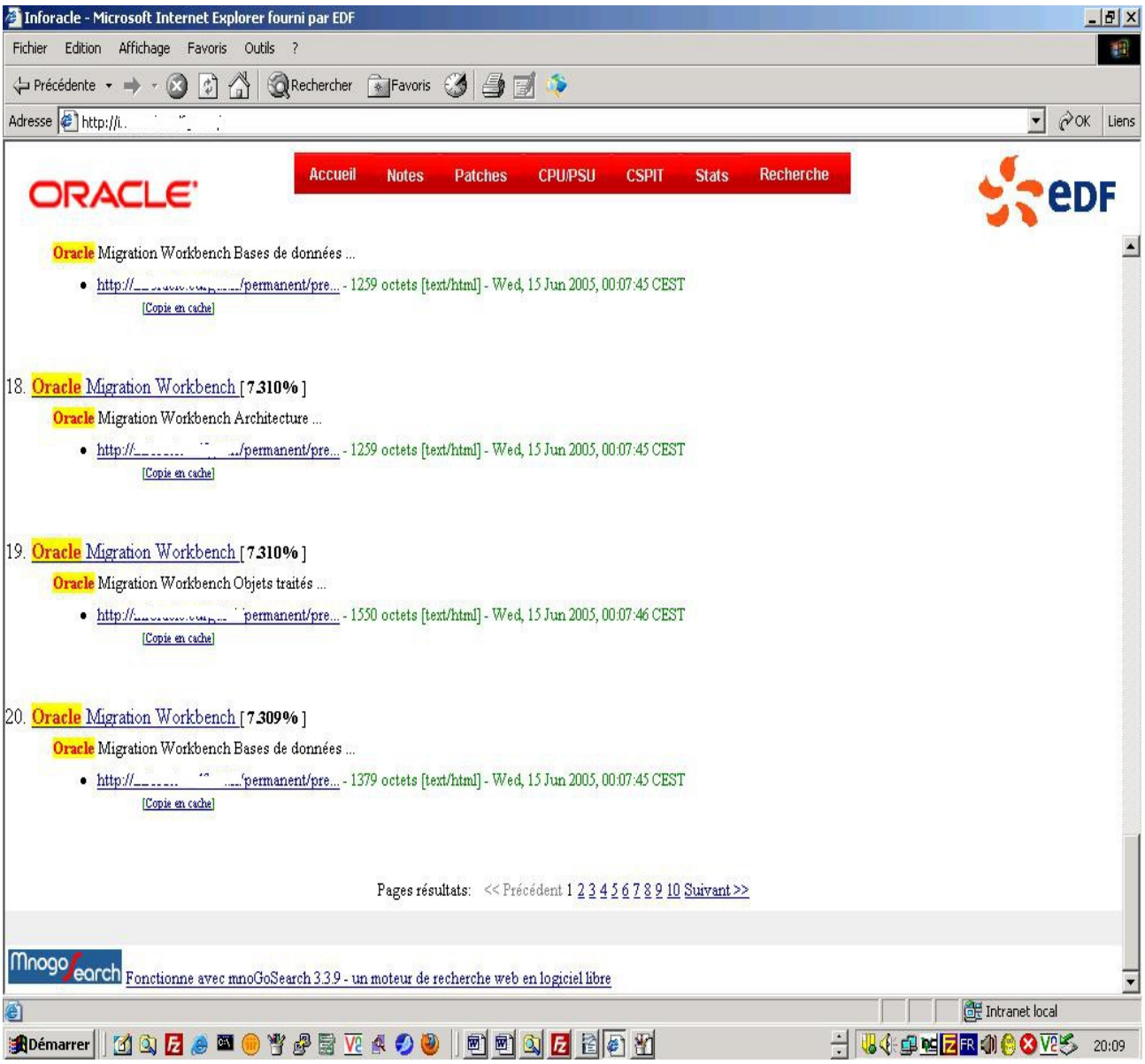

Figure 9 : capture d'écran avec numéro de navigation en bas de page

IV.1.5 Formulaire de recherche – aide à la recherche

L'appel du moteur de recherche à partir du site web indexé se fait à partir d'une page contenant un champ de recherche et auquel a été ajoutée une aide basique avec des exemples afin de permettre aux utilisateurs qui le souhaitent d'améliorer leur requête de

recherche avec les opérateurs booléens : ET logique, OU logique, NON logique et les parenthèses de groupement.

Voici une capture d'écran de cette page :

Figure 10 : capture d'écran de la page de recherche personnalisée avec aide sur les opérateurs booléens

IV.1.6 L'authentification et la sécurité dans mnoGoSearch

mnoGoSearch permet d'indexer des sites sécurisés par HTTPS lorsqu'il a été compilé avec une librairie apportant le support SSL comme OpenSSL. Il suffit juste de remplacer l'URL http:// par https:// dans le fichier indexer.conf.

De plus, ce moteur de recherche permet aussi d'indexer des sites web qui demandent une authentification basique. Dans ce cas il sera nécessaire de renseigner la variable AuthBasic avec l'identifiant et le mot de passe sous la forme login1:passwd1. mnoGoSearch peut indexer différents SGBD sous réserve que le moteur soit compilé avec les librairies clientes de ce ou ces SGBD ou alors avec UnixODBC et d'avoir un pilote ODBC pour Unix ou Linux.

Dans le cas des sites dont l'accès est restreint par identifiant et mot de passe, il faut commenter la ligne

« Section CachedCopy 25 64000 »

dans le fichier indexer.conf afin de retirer la possibilité à l'utilisateur d'accéder à la copie en cache du document sur le formulaire de recherche.

On a alors ce type de page de résultats sans le lien copie en cache :

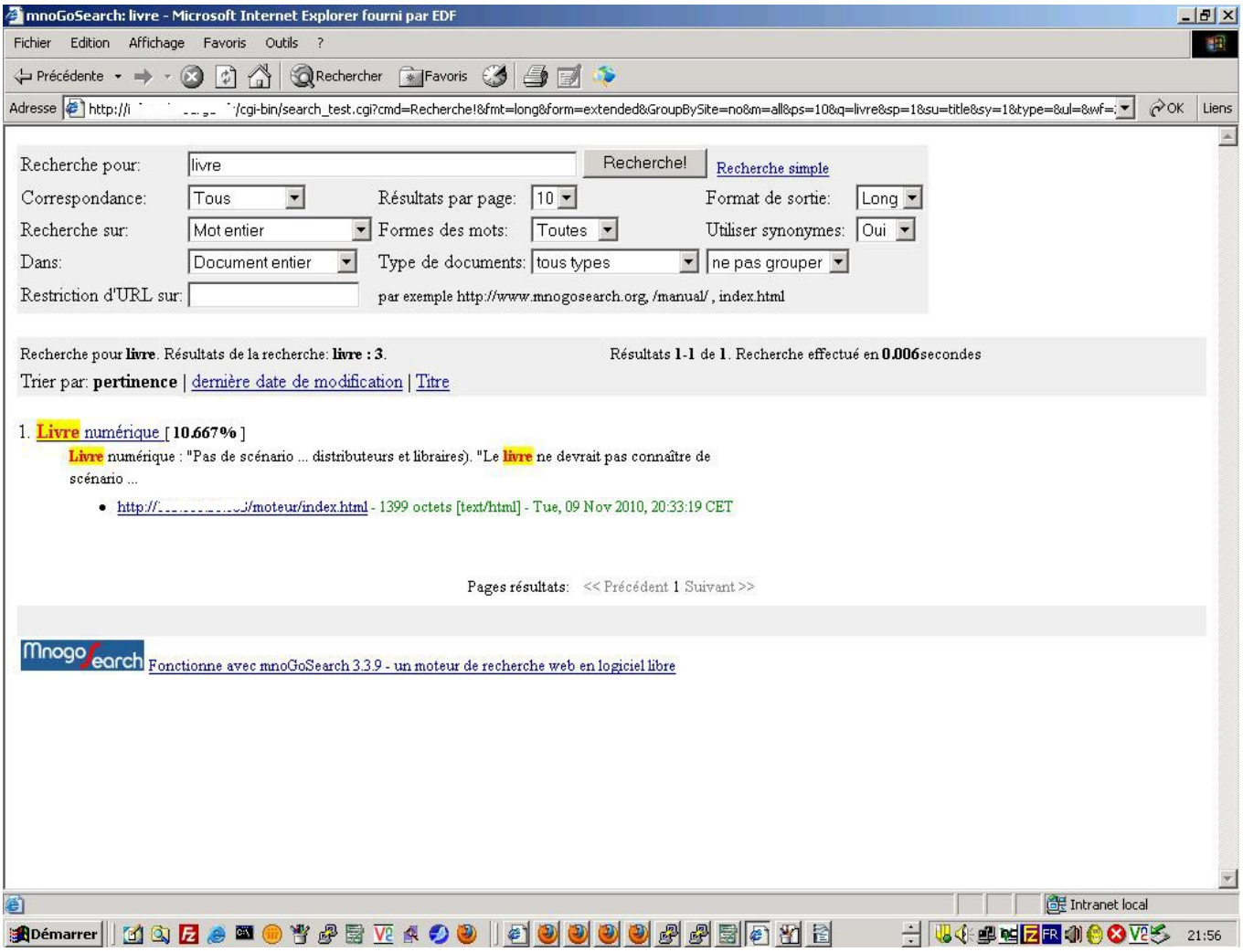

Figure 11 : capture d'écran de résultats sans cache de documents

De plus, lorsque l'on clique sur le lien d'un site sécurisé, alors une page blanche avec une mire d'authentification apparaît et nous demande l'identifiant et le mot de passe pour accéder au site comme ci-dessous.

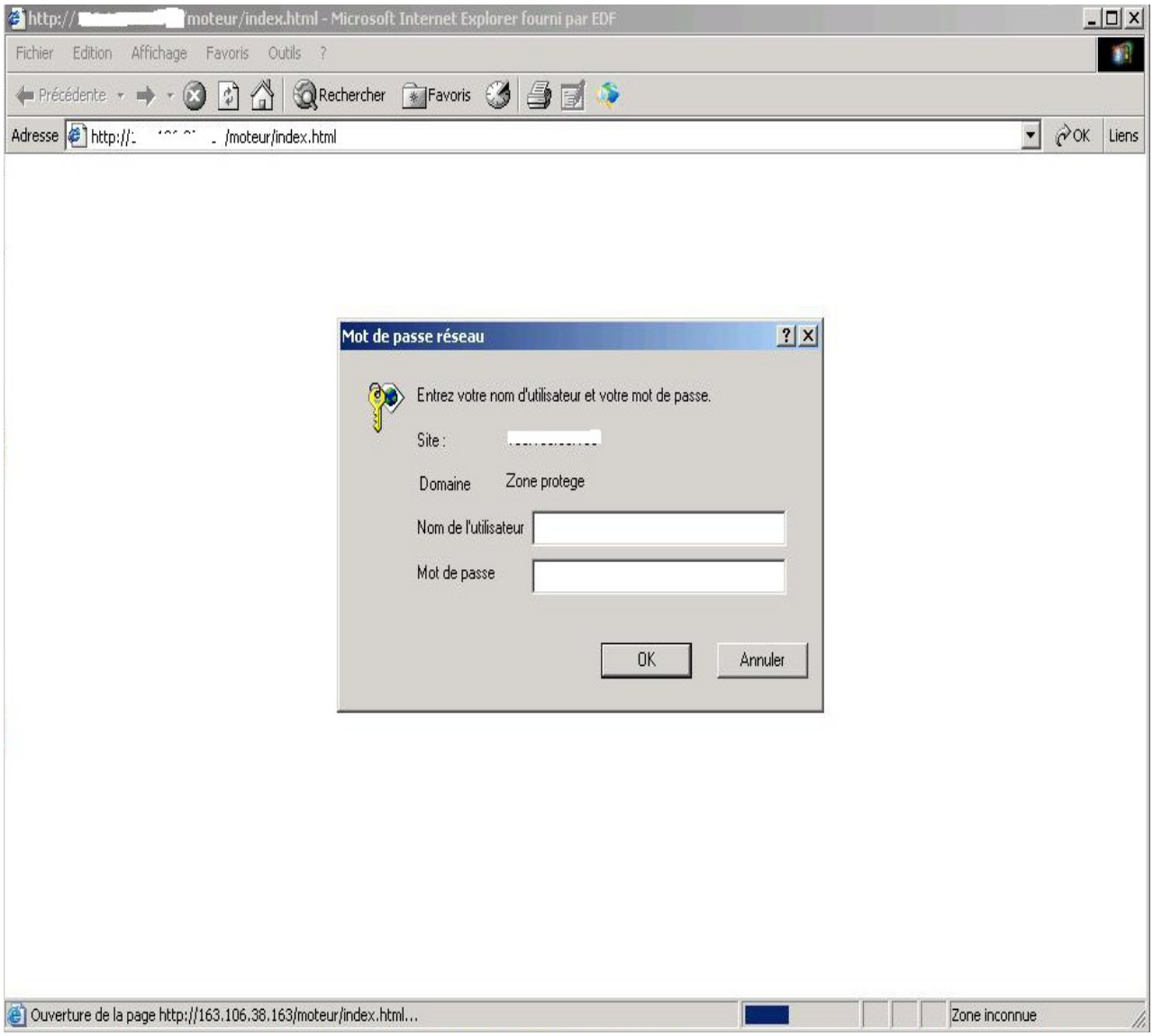

Figure 12 : capture d'écran d'un site sécurisé lors du clic sur un lien vers ce site dans les résultats de recherche

IV.1.7 Conclusion

Dans les différentes parties précédentes, nous avons vu dans le détail et par des images que mnoGoSearch est un moteur de recherche qui permet de répondre au besoin de recherche d'un utilisateur qui veut simplement faire une recherche avec quelques mots-clés et celui d'un utilisateur qui veut faire des recherches plus complexes en utilisant un ensemble d'options de recherche fournies par le moteur.

V Fonctionnalités avancées de recherche et aspect multi-langues

V.1 Normes d'encodage

Il a souvent été nécessaire dans le passé de faire un encodage de l'information sous une forme quelconque pour la transmettre. On peut citer le télégraphe de Chappe qui a permis le premier de transmettre des informations via des signaux visuels sur de longues distances, mais il y a aussi le code Morse qui lui a permis de transmettre des textes et messages de manière courante par radio ou ligne télégraphique pendant plus d'un siècle.

Dans l'informatique, il en va de même. Le premier encodage ou code pour transmettre l'information a été créé en 1963 et s'appelle l'ASCII (*American Standard Code for Information Interchange).*

Il faut préciser ici que le codage des caractères est un code associant un ensemble de caractères d'un jeu de caractères donné (appelé souvent page de code) avec quelque chose qui est un ensemble de nombre entiers dans le cas de l'informatique en général.

Voici à titre d'exemple la page de code ISO-8859-1 pour expliquer cette notion de page de code:

Les 191 caractères de ISO 8859-1 sont représentés sous forme de glyphes (œil) dans le tableau suivant. Les titres des lignes et des colonnes indiquent les codes hexadécimaux correspondant à chaque caractère, par exemple, le code hexadécimal de « L » est 4C, soit 01001100 en binaire ou 76 en décimal.

ISO/CEI 8859-1															
	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	*caractères de contrôle et divers non imprimables*															
1x																
2x		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x	*caractères de contrôle et divers non imprimables*															

9x															
Ax		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à		â	ã	ä	å	æ	ç	è	é	ê			í	î	ï
Fx	ð				ô			÷	ø	ù		û	ü	ý	þ	ÿ

Tableau 2 : Table des caractères de l'encodage ISO 8859-1

Source : http://fr.wikipedia.org/wiki/ISO_8859-1

mnoGoSearch intègre le support de nombreuses langues (environ 650) via de nombreux jeux de caractères, sur un ou plusieurs octets, utilisés par différents systèmes comme :

- sur MS Windows et Unix/Linux : page de code de l'ISO8859-1 à ISO8859-16, UTF-8
- sur MS Windows : page de code CP1250 à CP1258
- sur UNIX/Linux : ISO-2022-JP et EUC-JP pour le japonais, EUC-KR pour le coréen
- sur MacOS : MacCyrillic, MacRoman, MacCentralEurope

Dans le cas où mnoGoSearch aurait besoin de faire de la conversion entre jeux de caractères différents lors de l'indexation, il peut s'appuyer sur la librairie logicielle libiconv qui est dédié à ce type de conversion.

V.2 Implémentation dans mnoGoSearch

mnoGoSearch est un moteur de recherche multilingue. Il est capable de traiter des pages web et des documents bureautiques dans beaucoup de langues et encodés suivant différentes normes pour chaque langue. Cette capacité s'appuie sur 3 éléments : sur la recherche de variables définissant la langue dans les pages web, sur la définition de la langue et l'encodage à utiliser dans le fichier de configuration de mnoGoSearch, et sur la

détection automatique de jeux de caractères en utilisant une technique de catégorisation de texte basée sur N-Gram à partir de fichiers de données fournis avec le logiciel.

Dans les pages web en HTML, la langue de la page est indiquée parfois grâce au code HTML lang.

Exemple : lang="fr" pour le français.

V.3 Le segmenteur

Les langues de différents pays en Asie utilisent des idéogrammes et ne possèdent donc pas d'espace entre les caractères permettant de déterminer un mot. En chinois par exemple, il n'y a qu'un caractère représentant un rond et qui est un peu plus grand que le point utilisé en français pour indiquer la fin de la phrase. En conséquence une méthode permettant de définir que 2 ou plusieurs caractères forment un même mot est très utile pour faire des recherches efficaces. Pour résoudre ce problème, mnoGoSearch a implémenté un segmenteur qui permet de trouver des mots nécessitant plusieurs idéogrammes.

Ce support est fourni par les librairies Chasen ou Mecab pour le Japonais, et par des listes de caractères contenus dans un fichier texte pour le chinois simplifié, le chinois traditionnel et le thaï.

Pour une liste en chinois simplifié, on aura par exemple les 2 caractères :

电脑 diàn nǎo qui forment le mot qui veut dire ordinateur.

Et dans le fichier mandarin.freq à charger par mnoGoSearch pour l'indexation afin de lui indiquer qu'un ou plusieurs caractères forment un même mot, nous avons la ligne

692 电脑 dian4nao3

qui correspond à notre exemple précédent.

V.4 La liste des mots vides

L'indexation de l'ensemble des mots de nombreux documents peut générer des fichiers d'index dont la taille est importante. Afin de réduire cette taille, mnoGoSearch fournit environ une vingtaine de fichiers de liste de mots-vides pour différentes langues.

Ces fichiers sont à charger dans le fichier de paramétrage de l'indexation et dans le formulaire de recherche. Dans ce dernier, il est nécessaire de le charger pour indiquer à l'utilisateur dans le formulaire de résultats que certains des mots-clés qu'il a utilisés pour sa recherche sont des mots vides et donc qu'il n'aura pas de résultats en retour sur les documents contenant uniquement ceux-ci.

Il est à noter qu'il est souhaitable de faire une indexation dédiée à une langue pour un site web qui en contient plusieurs afin de ne charger qu'un seul fichier de mots-vides à la fois sinon il y aura des problèmes. Par exemple : si je charge en même temps le fichier des mots-vides français et anglais, le mot car" sera dans la liste des mots-vides du français mais correspond au mot voiture en anglais qui ne sera alors pas indexé.

V.5 Les fichiers des synonymes

Dans les recherches, il est parfois utile d'utiliser un fichier de synonymes pour trouver des documents avec des mots-clés auxquels on n'aurait pas pensé. mnoGoSearch fournit plusieurs fichiers de synonymes pour plusieurs langues dont le français et l'anglais. Ce fichier de synonymes est à charger pour la langue choisie dans le formulaire de recherche.

Lorsque l'on lancera une recherche avec le mot-clé voiture", le moteur de recherche indiquera aussi les documents qui contiennent le mot-clé automobile".

V.6 Identification des langues des documents indexés

Par défaut, mnoGoSearch se base sur la valeur des variables « charset » ou « encoding » dans l'en-tête HTTP de la page web » à indexer pour déterminer le jeu de caractères utilisé. Dans le cas contraire, il utilise un mécanisme de détection automatique de la langue et du jeu de caractères utilisés. A l'heure actuelle, il est capable de reconnaitre plus de 100 combinaisons diverses de jeux de caractères et de langues. Cette détection se fait grâce à une technique de catégorisation du texte basé sur la méthode `N-Gram`. Pour ce support, mnoGoSearch fournit des fichiers de carte de langue (*language map)* pour chaque paire de langue-jeu de caractères.

S'il manque un fichier qui ne correspond pas à la langue à indexer, alors il est possible de créer un nouveau fichier de carte de langue en utilisant l'utilitaire mguesser.

Ce dernier prend en entrée un fichier texte contenant des textes sur différents sujets (sports, politique, économie, etc, science) et fournit en sortie une nouvelle carte de langue. J'ai utilisé cet outil pour créer la carte langue pour le Hindi utilisé en Inde en récupérant des textes sur des sujets divers grâce à leur URL en anglais sur le site web d'informations générales de la BBC.

V.7 Formulaire de recherche multilingue

mnoGoSearch est fourni avec un formulaire de recherche en anglais qui permet des recherches simples sans utilisation autre que des mots-clés et des recherches avancées via un formulaire comportant de nombreuses options sélectionnables via des listes déroulantes. De nombreux utilisateurs non anglophones ne lisent pas l'anglais ou le comprennent de manière basique. J'ai donc traduit ce formulaire en français parce qu'il est indispensable lorsqu'un site web est dans une langue que le formulaire de recherche et la mise en forme des résultats de recherche se fassent dans la même langue que le site web afin de ne pas décourager l'utilisateur avec ce type de désagrément.

V.8 Utilisation de la fonction de lemmatisation avec les dictionnaires et les affixes d'Ispell

Dans les recherches, il est parfois utile d'utiliser Ispell pour lemmatiser les mots-clés et trouver ensuite des documents avec des mots-clés auxquels on n'aurait pas pensé. mnoGoSearch ne fournit pas les fichiers Ispell pour une langue donnée, il faut les télécharger à part sur Internet. Ces fichiers Ispell sont à charger pour la langue choisie dans le formulaire de recherche. Lorsque que mnoGoSearch est utilisé avec le support Ispell, tous les mots sont normalisés. Cette technique permet de trouver différentes formes grammaticales des mêmes mots. Durant l'indexation, tous les mots sont stockés « tel quel » dans la base de données. Lors de la recherche de l'utilisateur, toutes les formes du mot-clé recherché sont trouvées dans les documents existants indexés et remontés à l'utilisateur.

Lorsque l’on lancera une recherche avec le mot-clé test”, le moteur de recherche indiquera aussi les documents qui contiennent le mot-clé tests”.

VI Industrialisation du moteur de recherche retenu

L'industrialisation est un processus que je vais d'abord définir, puis expliquer de manière globale dans le contexte de ce mémoire et enfin je vais détailler l'ensemble du travail réalisé sur ce sujet.

Les objectifs principaux de l'industrialisation sont de réduire les coûts, les incidents et le temps pour réaliser une tâche.

Pour le Journal du Net, la définition générale de l'industrialisation est la suivante : « On peut définir l'industrialisation informatique comme une succession d'étapes qui conduisent à une meilleure gestion des ressources internes dans un contexte donné. Ce terme n'est pas à confondre avec l'automatisation des processus métiers, où il s'agit d'optimiser l'exécution des tâches quotidiennes par l'utilisation de l'outil informatique. En effet, dès qu'une direction parle d'industrialiser telle ou telle procédure, cela signifie que le processus a déjà été informatisé mais qu'il peut encore être amélioré. »

L'industrialisation passe souvent par la codification de bonnes pratiques existantes

(ayant été validées au sein de différentes entreprises ou dans un service de cette même entreprise) et par la formation de personnes, mais aussi par l'utilisation d'outils spécifiques pour le suivi ou la mise en œuvre ou par la création d'un service dédié sur une activité précise et dotée d'un responsable.

Il existe différents types d'industrialisation qui varient suivant le domaine d'application :

- Industrialisation logicielle

Certains indiquent que l'industrialisation logicielle couvre plusieurs aspects. En premier, nous avons l'industrialisation des logiciels. Lors de la création de logiciels, il est nécessaire de mettre en place une procédure permettant de développer dans de bonnes conditions, puis de tester le logiciel avant sa livraison au client. Il est donc nécessaire d'avoir des outils : pour gérer le code source, pour gérer et suivre les bogues rapportés et qui seront à corriger, pour effectuer des tests unitaires automatisés sur un ou des ensembles

de composants logiciels, pour faire et exécuter un plan de test de manière manuelle ou automatisée sur la partie fonctionnelle de l'application.

Lorsque le logiciel est considéré comme stable, il peut être déployé sur les serveurs ou les postes des utilisateurs. Pour s'y retrouver parmi ces nombreux logiciels utilisés, un service informatique peut mettre en place un suivi des licences achetées et utilisées sur le parc informatique. Des outils spécialisés permettront de faire un inventaire en temps réel en découvrant la liste des logiciels installés sur les postes. D'autres outils dédiés à la télédistribution permettront d'automatiser l'installation de logiciels ou de mises à jour. Le matériel informatique en tant qu'actif financier de l'entreprise sera suivi dans une solution de gestion comptable.

Lors de la création d'un logiciel, et de la gestion du projet qui y est rattaché, l'industrialisation passe par la mise en œuvre de cadriciels (frameworks) pour la création du code source suivant de bonnes pratiques, l'utilisation de méthodes comme la méthode Agile, le recours à la TRA (Tierce Recette Applicative) par une équipe à part pour s'assurer que le logiciel est conforme à ce qui est attendu. Il faut aussi gérer les calendriers et les étapes de développement avec une méthode comme PERT. La gestion des tâches, surtout dans les grands projets, devra se faire en attribuant des blocs logiciels autonomes à chaque équipe éloignée géographiquement afin de respecter la loi de Melvin Conway qui dit que « Les organisations qui conçoivent les systèmes ... sont contraintes de produire des modèles qui sont des copies de leur propre structure de communication ».

Tous ces logiciels et serveurs d'applications, web ou de base de données ne sont pas toujours sûrs et il arrive régulièrement qu'il soit nécessaire de corriger une faille de sécurité pouvant interrompre le service, corrompre, voire voler des données de la part d'une ou de personnes indélicates. Ces différents méfaits, et en particulier le vol de données, peuvent compromettre la réputation et la confiance dans une entreprise de manière irrémédiable. Aussi, afin d'éviter ces problèmes et de passer d'une attitude passive de défense corrective active à une attitude pro-active dans la gestion des menaces, il est souhaitable de créer un service dédié à la sécurité informatique qui sera chargé de la veille et de l'information des experts en charge des différents logiciels pour leur demander de prévoir un correctif ou une solution de contournement en cas de risques identifiés.

Avec un parc logiciel et applicatif important, un support informatique sera nécessaire pour traiter et débloquer rapidement les utilisateurs sur des incidents simples et

récurrents comme la perte du mot de passe mais aussi pour servir de point d'entrée afin de ré-aiguiller la demande de l'utilisateur vers une autre entité pouvant répondre à la requête. Cette cellule de support peut utiliser des bonnes pratiques de gestion d'incidents grâce à des logiciels dédiés mais aussi par une base de connaissances afin de répondre aux demandes les plus régulières tout en capitalisant la connaissance en cas de changement de personnes dans l'équipe.

- Industrialisation de l'infrastructure

Avec un parc informatique de plus en plus important au niveau des serveurs et au niveau des postes client, il est nécessaire d'appliquer différentes méthodes et pratiques pour utiliser, exploiter et avoir une bonne disponibilité sur l'ensemble des applications qui sont critiques pour le bon fonctionnement de l'entreprise ou de la structure les utilisant.

Au niveau des ordinateurs des utilisateurs et des serveurs, il est souhaitable d'avoir un parc le plus homogène possible dans le sens d'une même configuration matérielle avec quelques configurations types au niveau de l'utilisateur ou de l'utilisation. Cette standardisation permettra de créer une image du système d'exploitation, avec les pilotes matériels et les logiciels, qui sera installée sur une base importante d'ordinateurs de manière rapide, peu coûteuse et homogène.

Au niveau du réseau, pour industrialiser l'infrastructure et réduire le budget d'exploitation, il est souvent réalisé une cartographie des équipements, un plan de nommage standardisé, l'utilisation de matériel type et l'utilisation de règles réseau pour le paramétrage des équipements.

Avec de nombreux serveurs installés dans les entreprises et qui sont parfois peu utilisés tout en ayant un coût récurrent en électricité, en personnel d'exploitation et en hébergement pour le local, il est devenu de plus en plus courant de consolider, c'est à dire regrouper ces serveurs sur une seule machine puissante en les virtualisant. Cette virtualisation permet de réduire les coûts en adaptant le service aux besoins réels, au niveau matériel et du système de l'application, des services utilisés par les utilisateurs. De plus la virtualisation permet d'ajouter de la puissance de calcul ou de la mémoire supplémentaire à un système virtualisé de manière transparente et rapide sans changer de serveur ou d'effectuer des opérations d'ajout de composant matériel sur celui-ci. Dans le

cas de la virtualisation, l'industrialisation peut se faire en créant différents modèles de machine virtuelle dont le contenu se trouve dans et un ou plusieurs fichiers. Ce contenu est l'ensemble des fichiers du système d'exploitation, des pilotes pour le support du matériel, de logiciels divers et de paramétrages optimisés mais de manière générique pour un usage donné comme de la production informatique de type serveur web ou de base de données, du développement, etc... Grâce à ces différents modèles de machines virtuelles qui sont créés une fois pour toutes, il est possible de déployer plusieurs instances d'un même modèle (donc une copie de fichiers) rapidement en faisant juste le paramétrage réseau de la machine sans passer un temps très important pour installer un système d'exploitation et ensuite ces différents composants qui vont nécessiter de multiples démarrage du système.

Avec le rôle crucial de l'informatique dans l'entreprise, il n'est pas possible d'avoir des interruptions de service sur certaines applications, il est donc nécessaire de prévoir des architectures redondantes et interconnectées pour les serveurs, les équipements réseaux, l'alimentation électrique et le matériel de sauvegarde.

Pour les sites de commerce électronique utilisant des environnements web, l'industrialisation de leur infrastructure est une étape indispensable pour assurer la disponibilité du site web et des performances globales de leurs plateformes. Au commencement, il est possible de créer son site marchand sur un serveur qui hébergera l'ensemble des services ou serveurs spécialisés (web, applicatif, messagerie, base de données, etc...) afin d'avoir un coût d'investissement le plus bas possible. Mais dès que le site prendra de l'ampleur en ayant de plus en plus d'utilisateurs, il sera indispensable de mettre ces services ou serveurs spécialisés sur un serveur dédié. Par exemple, le serveur web en tant que frontal traitera les pages avec des données statiques, le serveur d'application traitera les données dynamiques. Quant au serveur de bases de données, il servira les données au serveur d'application qui se chargera de générer les pages dynamiques avec des données personnalisées en fonction de chaque utilisateur.

- Industrialisation des services informatiques

L'optimisation des services informatiques, tout comme les biens matériels, peut se faire grâce à des méthodes. Celles-ci permettent de préciser un service rendu de manière

objective en évaluant les délais moyens et le coût afin de pouvoir s'engager sur des niveaux de services appelés communément SLA (Service Level Agreement) ou des objectifs de service SLO (Service Level Objectives). La mesure objective du service rendu permet de faciliter les rapports entre prestataires de services et directions informatiques en s'appuyant sur des indicateurs tangibles et globaux et non quelques faits épars.

- Méthodes utilisées lors de la mise en place de procédures d'industrialisation

Parmi les méthodes les plus courantes, on trouve : CMMi, ITIL ou COBIT

CMMi (Capability Maturity Model + Integration) est un modèle de référence, un ensemble structuré de bonnes pratiques, destiné à appréhender, évaluer et améliorer les activités des entreprises d'ingénierie. Il permet d'appréhender et de mesurer la qualité des services rendus par les fournisseurs de logiciels informatiques. Il est utilisé par l'ensemble des acteurs pour évaluer et améliorer le développement de produits.

Le CobiT (Control Objectives for Information and related Technology) est un outil fédérateur qui permet d'instaurer un langage commun pour parler de la Gouvernance des systèmes d'information.

ITIL (Information Technology Infrastructure Library pour « Bibliothèque pour l'infrastructure des technologies de l'information ») est un ensemble d'ouvrages recensant les bonnes pratiques (« best practices ») pour le management du système d'information. L'utilisation d'ITIL permet d'avoir une qualité de service homogène et une démarche pro-active de réduction des incidents et problèmes récurrents provoquant des incidents de production qui ont des effets comme le coût humain en temps et en argent pour l'équipe d'exploitation, l'insatisfaction des utilisateurs, des coûts de ventes non réalisées sur un site web de vente en ligne par exemple.

Ces méthodes permettent à la direction informatique de définir des axes d'amélioration suivant une ou des échéances à préciser et ainsi d'obtenir les outils nécessaires pour les atteindre. Ces méthodes sont totalement agnostiques au niveau des outils à utiliser. Elles ne sont là que pour recommander des bonnes pratiques qui ont déjà été testées et qui devront être adaptées à l'entreprise cible.

De manière globale, on peut dire que l'industrialisation informatique est indispensable lorsque l'on passe d'une petite structure à une grosse structure. Il s'agit de passer d'un travail de manière artisanale à un travail de manière industrielle. Il y a cette notion de changement d'échelle qui change la manière de travailler avec comme objectif une réutilisation des méthodes, un transfert de compétences facilité entre personnes, une réduction des coûts.

VI.1 Définition de l'industrialisation

L'industrialisation informatique possède donc plusieurs définitions qui varient en fonction des personnes. Je vais donc définir ce que j'entends dans ce terme d'industrialisation et ce qu'il recouvre.

Dans le cadre de mon projet, l'industrialisation d'un logiciel est l'ensemble des opérations à réaliser afin de répondre aux objectifs suivants :

- adapter le logiciel au système d'exploitation et au matériel cible (par exemple, en compilant un logiciel et ses composants logiciels annexes pour un système d'exploitation comme Linux avec une version de distribution donnée et pour une famille de processeurs comme l'Intel x86)
- améliorer la sécurité standard du logiciel en désactivant des options ou en modifiant légèrement le comportement attendu pour éviter les attaques standards" sur un produit donné, (Par exemple, en retirant les numéros de versions affichés ou en obligeant un fonctionnement par un identifiant et mot de passe).
- standardiser le produit pour avoir une même logique que d'autres produits utilisés au sein d'une même entité comme une entreprise, (Par exemple, les binaires d'un logiciel et ses données vont se trouver dans des partitions de disque distinctes qui seront communes à plusieurs logiciels, ce choix permet ensuite de faciliter les sauvegardes en ne sauvegardant que le minimum des données nécessaires).
- réutiliser de bonnes pratiques antérieures, en reconduisant certains paramétrages logiciels qui ont été éprouvés sur de grosses applications avec beaucoup d'utilisateurs.

- réduire les coûts en diminuant le travail de l'exploitant, (Par exemple, en utilisant des scripts qui vont exécuter beaucoup de tâches qui prendraient du temps si elles étaient faites individuellement ou en factorisant" des fichiers de configuration, c'est à dire en appelant un autre fichier avec des paramètres génériques dans un fichier de configuration contenant les paramètres spécifiques à une application ou un site web).
- réduire les incidents en utilisant des méthodes d'exploitation, des configurations qui sont maitrisées et sûres. (Par exemple, en utilisant les paramètres optimisés du garbage collector" dans la JVM Java afin de réduire l'impact potentiel d'une fuite mémoire entrainant une indisponibilité de l'application par manque de mémoire disponible).

VI.2 Démarches effectuées lors de l'industrialisation

Dans le cadre du projet de mon mémoire, j'ai rédigé dans un premier temps un cahier des charges regroupant les spécifications obligatoires et optionnelles du moteur de recherche à retenir dans le cadre d'une utilisation chez EDF.

Après avoir déterminé les éléments de choix de ce logiciel, j'ai réalisé un inventaire de l'ensemble des logiciels de moteur de recherche en « open source » disponibles sur Internet et ai retenu les finalistes pour une étude détaillée avec la méthode QSOS en fonction de leur dernière date de mise à jour, du support offert et du support du protocole HTTPS. L'étude QSOS, avec l'identification de l'ensemble des fonctionnalités des moteurs à évaluer en fonction du cahier des charges, m'a permis de choisir le moteur de recherche mnoGoSearch.

Ensuite, il m'a fallu me familiariser avec mnoGoSearch dans son fonctionnement, sa configuration et ses possibilités. Après cette étape, j'ai réalisé la compilation de mnoGoSearch et de ses composants logiciels annexes ou externes qui n'existent que sous forme de sources. Cette opération s'est effectuée en fonction des répertoires normalisés et des plateformes serveurs disponibles à EDF.

Puis, j'ai fait un travail, par itérations et essais et tests, d'identification des paramètres à sécuriser ou à modifier pour les adapter au contexte EDF ou faciliter leur paramétrage. Cette étape m'a permis de séparer le fichier d'indexation standard en 2

fichiers, l'un contenant les paramètres génériques pour une utilisation avec une langue à indexer d'Europe occidentale et l'autre avec les paramètres spécifiques à un site web ou une langue.

Afin de faciliter l'exploitation de mnoGoSearch, j'ai créé différents scripts (shell Unix/Linux) pour créer ou supprimer la base de données contenant les index, lancer l'indexation, connaître le nombre d'URLs indexés ou en erreurs d'une même collection d'index regroupés dans la même base de données. Le formulaire de recherche fourni avec mnoGoSearch étant en anglais uniquement, je l'ai localisé et traduit en français afin de faciliter sa prise en main par les utilisateurs francophones.

Après avoir assemblé l'ensemble de ces briques logicielles, il m'a été nécessaire de tester ces composants en rédigeant un plan de test qui m'a permis de m 'assurer que les principales fonctions (indexation, parseurs internes de mnoGoSearch, parseurs externes bureautiques, stockage dans une base de données des index, recherche en mode simple et avancée) étaient opérationnelles.

VI.3 Opérations réalisées lors de l'industrialisation

Je vais ensuite décrire l'industrialisation sur les différents composants logiciels suivant : le moteur de recherche, les parseurs, le fichier de configuration principal, le formulaire de recherche et les scripts.

VI.3.1 Industrialisation réalisée sur le moteur.

Au début de mon travail sur le moteur mnoGoSearch, j'ai essayé d'abord de le faire fonctionner a minima, c'est à dire de valider le fonctionnement de la chaine récupération des données -> indexation -> recherche" afin de progresser par étapes dans la maitrise du produit.

mnoGoSearch étant un logiciel développé en langage C, il nécessite donc d'être compilé pour un environnement cible. J'ai donc vérifié les options du fichier (en script shell) configure" des sources du moteur qui sont à passer à l'environnement de développement avant la compilation. Cette dernière opération m'a permis de décider

d'activer la compilation avec les options disponibles pour les différents SGBD (Systèmes de Gestion de Base de Données) suivants :

- **Mysql** : une base Mysql sera utilisée pour stocker les index d'autant plus que ce SGBD ne nécessite pas de licence logicielle de l'éditeur, il a de bonnes performances en lecture/écriture sur le traitement des données, il est le moteur de base de données principal pour le développement de nouvelles fonctionnalités dans mnoGoSearch et, au final, il a donc un support optimisé voire excellent avec ce moteur de recherche.
- **Oracle** : mnoGoSearch a été compilé avec les librairies clientes d'Oracle DB afin de permettre l'accès et donc l'indexation de données stockées dans des bases Oracle. Le support de Mysql et Oracle est important dans mnoGoSearch, puisque ce sont les 2 SGBD au référentiel pour toute l'entreprise.
- **Sqlite** : mnoGoSearch a été compilé avec les librairies clientes de Sqlite afin d'offrir la possibilité de stocker des index ayant une faible volumétrie lorsque l'administrateur de mnoGoSearch n'a pas de compétences pour administrer un serveur Mysql ou Oracle. Ici la base de données est donc un simple fichier.
- **UnixODBC**: mnoGoSearch a été compilé avec les librairies du connecteur UnixODBC afin de permettre un accès à toutes les bases de données ayant un pilote ODBC.

  De plus cette option autorisera le support de nouvelles versions des SGBD Oracle, Mysql, SQlite dans le futur si le produit n'est pas recompilé avec les nouvelles versions des librairies clientes.

J'ai compilé aussi mnoGoSearch avec d'autres options qui apportent chacune une fonctionnalité supplémentaire au moteur. Voici ces options :

- **zlib** : cette librairie permet de supporter de l'encodage du contenu HTTP
- **readline** : cette librairie permet d'ajouter le support du moniteur SQL
- **openssl** : cette librairie permet d'ajouter le support du protocole SSL qui est indispensable pour accéder aux données de sites web en HTTPS

- **jeux de caractères** : le support de tous les jeux de caractères pour les caractères des langues asiatiques (Chine, Japon, Inde) est activé
- **chasen** : cette librairie permet d'ajouter le support d'un système morphologique pour le japonais.

VI.3.2 Industrialisation réalisée sur les parseurs de documents bureautiques.

L'ensemble de ces options permettent de compiler un moteur déjà très complet mais mnoGoSearch n'intègre que le strict minimum dans ces parseurs intégrés pour les types MIME de documents suivants : text/html(*.htm ou *.html), text/xml (*.xml), text/plain (*.txt) and audio/mpeg (*.mp3). En conséquence, il m'a fallu compiler plusieurs logiciels qui agissent en tant que parseurs externes et permettent à mnoGoSearch d'indexer d'autres types de documents. Pour tous les parseurs, le principe de fonctionnement est toujours le même. Lorsque mnoGoSearch détecte un fichier dont il ne supporte pas le type MIME, il vérifie dans son fichier d'indexation si un parseur est associé à ce type MIME et transmet alors le fichier à indexer à ce parseur et en récupère la sortie texte, via le flux de sortie standard.

Les différents parseurs que j'ai compilés ou créés sont récapitulés dans la liste suivante :

- **Catdoc** est un utilitaire qui sait lire les fichiers Microsoft Word (*.doc), Excel (*.xls) et Powerpoint (*.ppt) au format binaire et en extrait uniquement le texte vers la sortie standard, c'est à dire l'écran de l'utilisateur. J'ai compilé la dernière version de ce logiciel.

  Mais j'ai aussi découvert que ce logiciel détectait le numéro de la page de code utilisé pour chaque fichier (http://msdn.microsoft.com/en-us/goglobal/bb964654) et ensuite utilisait un fichier fourni par l'organisation UNICODE qui permet la correspondance entre le numéro du standard UNICODE et la lettre ou le caractère à afficher (ftp://ftp.unicode.org/Public/MAPPINGS/). En examinant le code source de ce logiciel et les fichiers de correspondance fournis par l'organisation UNICODE, je me suis aperçu que certains jeux de caractères ou langues n'étaient pas supportés et j'ai donc patché" le code source de catdoc pour

ajouter cette possibilité et mis à jour les fichiers de correspondance UNICODE avec leur dernière version.

- **Unrtf** permet de convertir des fichiers *.rtf au format plain texte ou HTML. J'ai compilé ce logiciel sans modifications.
- **Xpdf** est un logiciel qui a été créé pour lire les documents au format PDF d'Adobe dans une interface graphique. Ce logiciel contient un utilitaire appelé pdftotext qui réalise le même travail que ceux de Catdoc : Il lit un ou plusieurs fichiers au format binaire Portable Document (*.pdf) et en extrait uniquement le texte vers la sortie standard, c'est-à-dire l'écran de l'utilisateur.

  Mon travail sur ce logiciel a constitué d'abord à le compiler, mais par défaut il ne supporte que les langues et les jeux de caractères pour l'Europe Occidentale. J'ai donc fait un travail de maximisation des possibilités du logiciel par défaut, au niveau des langues supportées, en récupérant les fichiers de correspondance (mappages) des jeux de caractères en fonction de la langue et les fontes TTF sur différents sites Internet afin de les stocker dans un sous-répertoire des binaires du logiciel Xpdf. Mais pour faciliter l'utilisation du parseur xpdf dans ces différentes langues ajoutées, j'ai créé un fichier de configuration prêt à l'emploi pour chaque langue qui sera à passer en paramètre à xpdf pour les langues non supportées par défaut.

  Parmi ces dernières langues, on a : les langues d'Europe Centrale, le Cyrilique pour les langues de la **C**ommunauté des **E**tats **I**ndépendants (ex-URSS), l'arabe, le grec moderne, l'hébreu, le thaï, le turc, le chinois en écriture simplifiée et traditionnelle, le japonais et le coréen.
- **Ghostscript** permet de convertir des fichiers *.ps (PostScript) au format plein texte. J'ai compilé ce logiciel sans modifications si ce n'est l'ajout des fontes TTF nécessaires à son fonctionnement.
- **sxwplain** est un script qui permet de convertir des fichiers *.sxw (StarOffice Writer) au format plain texte. Ce script s'appuyant sur différents utilitaires en ligne de commande, j'ai dû compiler les logiciels : **unzip** (décompression de fichiers zip, le sxw est un fichier zip renommé avec une

autre extension), **sed** pour le traitement du fichier xml par des expressions régulières afin d'en extraire le texte vu par l'utilisateur.

- **odtplain** est un script qui permet de convertir des fichiers *.odt (OpenOffice Writer) au format plein texte. J'ai créé ce script en l'adaptant du précédent sxwplain.
- **docxplain** est un script qui permet de convertir des fichiers *.docx (versions Microsoft Word récentes) au format plain texte. J'ai créé ce script en l'adaptant du précédent sxwplain.
- **xlhtml** permet de convertir des fichiers Microsoft Word (*.doc) ou Excel (*.xls) ou PowerPoint (*ppt) au format HTML pour les indexer ensuite par mnoGoSearch. J'ai compilé ce logiciel sans modifications. Ce logiciel semble redondant avec catdoc, mais il n'en est rien. Le parseur de fichier *.ppt de xlhtml est nécessaire parce que le parseur catppt du logiciel catdoc ne fonctionne pas correctement avec les fichiers *.ppt PowerPoint 97 à 2003, il renvoie une ligne « Thème Office » au lieu du contenu texte du fichier *.ppt et cela quelle que soit la version de catdoc utilisée 0.94.2 ou 0.94.4.
- **poppler** est une librairie logicielle libre qui est utilisée pour traiter les documents PDF par les logiciels avec interface graphique afin de visualiser ce type de document dans les interfaces graphiques KDE et Gnome. Ce logiciel contient un utilitaire qui réalise le même travail que celui de Xpdf: Il lit un ou plusieurs fichiers au format binaire Portable Document (*.pdf) et en extrait uniquement le texte vers la sortie standard, c'est à dire l'écran de l'utilisateur sous format HTML et non en texte brut.
- **Libpwd** est une librairie logicielle libre qui est utilisée pour traiter les documents WordPerfect Windows (*.wpd) et est utilisée par les logiciels bureautiques comme LibreOffice pour visualiser ou importer ce type de document. Ce logiciel contient deux utilitaires qui permettent d'extraire le texte du document sous forme de texte brut ou au format HTML vers la sortie standard, c'est à dire l'écran de l'utilisateur.
- **Libwps** est une librairie comme libpwd et fait la même chose qu'elle mais pour les documents MS Works (*.wps).

- **SofficeToHTML** permet de convertir des fichiers *.sxw (StarOffice/OpenOffice 1.x Writer) au format HTML. Comme ce script Perl ne permettait pas de réaliser une extraction avec un format HTML correct et de plus sur la sortie standard c'est à dire l'écran et non dans un fichier comme dans le logiciel d'origine, j'ai réalisé différentes modifications dans le script pour corriger ces problèmes gênants.
- **Odftools** permet de convertir des documents OpenOffice/Libreoffice au format/norme ODF (Open Document Format) (*.odt) au format HTML pour les indexer ensuite par mnoGoSearch.
- **Docx2txt** permet de convertir des documents Microsoft Word 2007 et supérieur (*.docx) au format plain texte.
- **Pptx2txt** est un script Perl qui permet de convertir des documents Microsoft PowerPoint 2007 et supérieur (*.pptx) au format plain texte.
- **Xlsx2csv** permet de convertir des documents Microsoft Excel 2007 et supérieur (*.xlxx) au format plain texte.

L'ensemble de ces logiciels permettent de répondre à l'indexation des documents bureautiques les plus courants que ce soit dans le monde MS Windows ou celui d'Unix/GNU Linux.

VI.3.3 Industrialisation réalisée sur le fichier de configuration de l'indexation du moteur

Le fichier « indexer.conf » qui sert à définir la presque totalité des paramètres à transmettre au binaire « indexer » est très long. En conséquence et afin de faciliter l'exploitation, j'ai scindé ce fichier en 2 parties. Le premier fichier, nommé « common-indexer-latin1.conf », contient les paramètres génériques pour l'indexation de documents utilisant l'encodage pour les langues d'Europe occidentale (CP1252/ISO-8859-1). Il est dans la partie Annexes de ce mémoire. Le second fichier garde son nom d'origine « indexer.conf ». Ce dernier charge le fichier common-indexer-latin1.conf et ensuite définit différents paramètres dont certains sont repris dans l'autre fichier mais sous leur forme décommentée c'est à dire que les paramètres deviennent actifs et seront utilisés. Cet ensemble de variables dans le paramétrage est spécifique à la source à indexer.

Fichier indexer.conf :

```
# Fichier de configuration de l'indexation
# d'un site avec une langue d'europe occidentale

# chargement des parametres generiques
# pour un site avec une langue d'europe occidentale
include /opt/mnogosearch_3.3.14/etc/common-indexer-latin1.conf

### Les variables DBAddr, Server non commentees ci-dessous
### sont les variables minimales a configurer pour un site web

# Parametrage de l'acces a base de donnee de stockage des index
#DBAddr <DBType>:[//[DBUser[:DBPass]@]DBHost[:DBPort]]/DBName/[?dbmode=mode]
#DBAddr    mysql://foo:bar@localhost/mnoGoSearch/?dbmode=single
# syntaxe DBAddr  mysql://login:mot_de_passe@IP_ou_nom_hote/nom_de_la_base/?
dbmode=single
# Le parametre ps=yes necessite mnoGoSearch >= 3.3.8 ET Mysql >= 4.1.x
DBAddr     mysql://mnoGoSearch:mnoGoSearch@localhost:3306/mnoGoSearch/?
dbmode=blob&ps=yes

# Repertoire de cache des resultats de recherche
VarDir /var/mnogosearch_3.3.14/cache

# L.indexeur detectera le meme document a differents endroits
# et indexera qu'un seul document pour les diferentes URL.
# Permet de reduire la taille des index malgre un ralentissement de l.indexation
# Valeur par defaut "no".
DetectClones yes

# Definition du jeu de caracteres qui sera utilisee pour stocker
# les donnees dans la base de donnee. Par defaut iso-8859-1
#LocalCharset windows-1252
LocalCharset iso-8859-1

# Utilisation de cookies pour les sessions
# Par defaut : non
#UseCookie no

# Commentez cette ligne si vous ne voulez pas stocker des "copies de documents en
cache (cached copies)"
# pour gerer des extraits intelligents de documents au moment de la recherche.
# ATTENTION : Si vous indexer un site securise par identifiant etr mot de passe,
# il est conseille de commenter ce parametre pour des raisons de securite.
# N'oubliez pas de conserver la section "Charset" active si vous utilisez
# des copies de documents en cache (cached copies).
#
# NOTE: 3.2.18 a une taille maximum pour CachedCopy, 32000 pour Ibase et
# 15000 pour Mimer. Les autres bases de donnees n'ont pas ces limites.
# Si l'indexeur echoue avec le message d'erreur 'string too long' alors reduisez
# cette valeur. Cela sera corrige dans les versions futures.
#
Section CachedCopy              25 64000

# Authentification HTTP basique
# par login et mot de passe et sur un repertoire precis
#AuthBasic login1:passwd1
#Server http://my.server.com/my/secure/directory1/

# Indiquer URL du site a indexer ici
#Server http://www.monsite.com
Server http://localhost

# Indiquer URL a exclure ici
#Disallow *.toto

# Indiquer URL particuliere du site andexer ici:
#URL http://www.monsite.com/plan.htm

##### la partie ci-dessous est a modifier en cas d'utilisation
##### d'une autre langue que le francais et l'anglais
##### En cas de pages web avec plusieurs langues
```

```
##### Il faut charger les fichiers des langues necessaires en meme temps

# Chargement du fichier des mots-vides pour reduire taille des index
# a configurer suivant la langue
# Liste des autres langues dans stopwords.conf

# Francais
StopwordFile /opt/mnogosearch_3.3.14/etc/stopwords/fr.sl
# Anglais
StopwordFile /opt/mnogosearch_3.3.14/etc/stopwords/en.sl
# Allemand
# StopwordFile /opt/mnogosearch_3.3.14/etc/stopwords/de.sl


# Chargement de la carte des langues pour trouver
# le jeu de caracteres ou la langue
# Liste des autres langues dans langmap.conf

# Francais
LangMapFile /opt/mnogosearch_3.3.14/etc/langmap/fr.latin1.lm
LangMapFile /opt/mnogosearch_3.3.14/etc/langmap/fr.latin1.bible.lm
# Anglais
LangMapFile /opt/mnogosearch_3.3.14/etc/langmap/en.ascii.lm
# Allemand
#LangMapFile /opt/mnogosearch_3.3.14/etc/langmap/de.latin1.lm
#LangMapFile /opt/mnogosearch_3.3.14/etc/langmap/de.latin1.bible.lm
```

VI.3.4 Industrialisation du formulaire de recherche search.htm

Le formulaire de recherche et d'affichage des résultats de recherche est fourni uniquement en anglais. Anticipant la difficulté, pour des utilisateurs francophones, d'utiliser ce formulaire surtout en mode recherche avancée avec des options en anglais et ensuite avec la page des résultats de recherche en anglais, j'ai donc traduit toutes les chaines de texte du formulaire en français. Dans une deuxième étape, pour généraliser mon approche, j'ai souhaité utiliser une nouvelle option du moteur : ReplaceVar" qui remplace une variable par une chaine texte afin de rendre le formulaire de recherche multilingue. J'ai donc dû refaire le travail en remplaçant toutes les chaines texte que j'avais traduites en français par des variables que j'ai créées et nommées. Ensuite, j'ai constitué un fichier texte avec, pour chaque ligne, une correspondance entre une variable unique et le texte/la traduction correspondante. Il reste ensuite à indiquer le fichier avec les correspondances/la traduction souhaitée à charger au début du formulaire de recherche.

Pour améliorer les résultats de recherche, des fichiers Ispell pour connaitre les racines, des mots et des fichiers pour les synonymes ont été ajoutés. Les fichiers Ispell permettent de faire une recherche avec mnoGoSearch sur la racine du mot recherché et ses dérivés. Exemple : avec gazier, on recherche gazière, gazières, gaziers. Les fichiers Ispell ont été inclus pour les langues : français, anglais, allemand, italien, espagnol. Pour la recherche sur les synonymes, mnoGoSearch contient déjà des fichiers avec les synonymes pour l'anglais, le français, l'italien et le russe. Sur Internet, j'ai réussi à trouver des fichiers de synonymes pour l'allemand et un fichier plus complet pour l'italien. En faisant une

recherche sur voiture par exemple, ces fichiers permettent de trouver les autres résultats contenant auto, automobile.

VI.3.5 Création de scripts d'exploitation

Afin de faciliter l'exploitation du moteur de recherche après sa mise en production, j'ai créé 4 scripts dédiés à 4 tâches principales qui seront décrits brièvement par la suite. Chaque script porte un nom explicite qui permet d'identifier facilement l'objectif du script.

- script « creation-database-mnoGoSearch » :

```
/opt/mnogosearch_3.3.14/sbin/indexer -Ecreate -d /opt/mnogosearch_3.3.14/conf/indexer.conf
```

Ce script sert à créer les tables pour stocker les index dans la base de données indiquée dans le fichier « indexer.conf » et créée au préalable.

- script « suppression-database-mnoGoSearch »

```
/opt/mnogosearch_3.3.14/sbin/indexer -Edrop -d /opt/mnogosearch_3.3.14/conf/indexer.conf
```

Ce script sert à supprimer les tables stockant les index dans la base de données indiquée dans le fichier « indexer.conf » et existant au préalable.

- script « indexation-mnoGoSearch »

```
/opt/mnogosearch_3.3.14/sbin/indexer -N1 -d /opt/mnogosearch_3.3.14/conf/indexer.conf
/opt/mnogosearch_3.3.14/sbin/indexer -Eblob -d /opt/mnogosearch_3.3.14/conf/indexer.conf
```

La première ligne sert à « crawler » c'est à dire à récupérer les mots destinés à l'indexation d'un site web ou système de fichier en fonction du paramétrage qui inclut la base de données dans « indexer.conf » puis va stocker ces mots dans cette base de données.

Le paramètre -N sert à indiquer le nombre de threads utilisés par le processus indexer afin de traiter plusieurs indexations sur une même source de données en même temps.

La seconde ligne avec le paramètre -Blob crée les index nécessaires à la recherche dans la base de données.

- script « statistiques »

```
/opt/mnogosearch_3.3.14/sbin/indexer -Edrop -d /opt/mnogosearch_3.3.14/conf/indexer.conf
```

Ce script sert à afficher des statistiques sur les indexations comme : nombre de documents indexés, nombre de documents non trouvés, nombre de documents expirés, nombre de documents interdits ou dont le format n'est pas supporté.

VI.3.6 Mise à jour d'un script Perl de décompression d'archives compressées

mnoGoSearch ne permet pas de décompresser des archives compressés dans n'importe quel format. Or c'est un handicap parce que souvent de nombreux documents sont compressés au format *.zip sur les serveurs pour gagner de l'espace sur le disque et réduire le temps de téléchargement pour l'utilisateur. J'ai donc récupérer un vieux script Perl appelé udm-gateway.pl disponible dans la partie téléchargement sur le site web www.mnoGoSearch.org. Je l'ai ensuite mis à jour pour supporter de nouveaux formats de compression comme le *.xz et le *.7z et de nouveaux formats de documents bureautiques (*.docx, *.pptx, *.xlsx) comme les formats XML de la suite bureautique MS Office à partir de la version 2007. De plus, il m'a fallu compiler plusieurs logiciels qui agissent en tant que logiciels spécialisés pour décompresser un certain type d'archives. Le script udm-gateway.pl est appelé par mnoGoSearch en tant que parseur externe lorsqu'il rencontre un type MIME d'archive qui a été configuré dans le fichier de configuration indexer.conf de l'indexation.

VI.3.7 Bilan

Avant de réaliser mon industrialisation, j'ai réalisé un cahier des charges pour déterminer les fonctionnalités attendues pour le moteur de recherche souhaité et ensuite choisir le moteur le plus adapté au contexte de l'entreprise. Ce document m'a permis aussi de prévoir l'ajout de composants externes au moteur retenu pour atteindre un nombre plus important de fonctionnalités attendues.

Pour réaliser mon industrialisation, j'ai récupéré les documents de constitution de produits industrialisés à EDF et qui nécessitent une compilation des sources : Apache, PHP, Mysql comme il est nécessaire de le faire avec mnoGoSearch. J'ai pu m'inspirer de ces documentations pour reprendre la même méthodologie éprouvée et la même installation, afin de s'intégrer de manière aisée sur les serveurs de production installés avec un système d'exploitation personnalisé à EDF.

Après avoir choisi le moteur le plus adapté et m'être imprégné de la méthodologie d'industrialisation citée ci-dessus, j'ai créé des scripts de compilation comme ceux déjà existant pour la souche EDF pour Apache. Je me suis d'abord attaqué à la compilation de mnoGoSearch uniquement avec le SGBD Mysql. Après cette étape je me suis familiarisé avec la configuration, la prise en main et les tests de ce bloc de composants minimaux. Ensuite j'ai réalisé les mêmes opérations en y intégrant les parseurs externes indispensables pour les documents bureautiques de la suite Office de Microsoft (Excel, Word, PowerPoint). Après cette intégration, j'ai ajouté de nouveaux connecteurs pour l'indexation et le stockage des index dans les SGBD Sqlite, Oracle DB et d'autres SGBD accessibles via un pilote ODBC. Voyant l'évolution des suites bureautiques, j'ai ajouté des parseurs pour les documents RTF, au format OpenOffice et les nouveaux formats utilisant XML dans MS Office. Je me suis occupé ensuite de l'internationalisation du formulaire de recherche et du support de jeux de caractères de langues non occidentales dans mnoGoSearch et les parseurs de documents.

On voit donc ici que l'industrialisation est une démarche itérative en allant du plus simple au plus compliqué et en maîtrisant les briques logicielles une à une. Je pense que si je devais le refaire, je le referais de la même façon et que si une autre personne avait dû faire le même travail, elle aurait plus au moins appliqué la même méthode que moi en se basant sur les documentations EDF déjà existantes de constitution de souches par

compilation. En revanche, il est clair que ce travail nécessite un intérêt pour le domaine des moteurs de recherche et donc d'avoir des compétences générales variées sur leur mode de fonctionnement, sur les protocoles réseau, sur l'encodage des caractères, sur les formats de documents bureautiques, les bases de données, les serveurs web, les pages en HTML, etc ... Le moteur de recherche peut, s'il n'est pas industrialisé, être un produit complexe à mettre en œuvre et être réservé à des experts du domaine parce que c'est un logiciel spécialisé par rapport aux besoins courants dans une équipe informatique.

mnoGoSearch est très flexible en permettant d'ajouter des parseurs externes que l'on peut récupérer sur Internet en utilisant la sortie texte de logiciel courant ou en en créant de toutes pièces suivant le besoin de l'utilisateur. Au final, cette industrialisation a permis d'avoir un logiciel de moteur de recherche très complet grâce à l'assemblage de tous ces composants logiciels externes.

Je considère avoir réussi l'industrialisation presque à 100 %. Le moteur mnoGoSearch industrialisé avec ces composants externes propose un bon produit répondant aux besoins de base et avancés d'une fonction de recherche sur un site web ou dans une base de données. Il ne lui manque qu'un script qui pourrait créer les répertoires, mettre en place les fichiers et scripts, et en plus paramétrer certaines variables dans le fichier de configuration de l'indexation. A cause du temps important nécessaire à la création de ce script et ensuite des nombreux tests à réaliser pour deux systèmes d'exploitation au référentiel que sont Linux et Solaris, je ne l'ai pas réalisé. Ces derniers systèmes d'exploitation ont, de plus, des commandes ou des comportements de commandes pour les scripts, qui sont parfois spécifiques au système. Malgré l'absence de ce script de création de répertoires et de mise en place des fichiers d'une instance du moteur de recherche, la mise en œuvre de l'indexation et de la recherche sur un site web est assez facile en suivant la procédure décrite dans la documentation.

L'intérêt de la démarche d'industrialisation est de fournir d'une manière générale, un logiciel adapté à un contexte d'utilisation et pré-configuré de manière automatique, ou pouvant être pré-configuré facilement par un simple utilisateur comme un exploitant sans aucune connaissance du produit et en suivant des directives ou en utilisant les informations que lui donneraient le projet ou l'intégrateur de l'application. Il s'agit donc ici d'un gain de temps et d'argent au niveau formation et, en particulier, pour les logiciels où il y a peu de documentation en ligne.

L'ensemble des étapes de l'industrialisation (compilation, création de scripts et de fichiers de configuration) est détaillé dans la documentation "Dossier d'installation du Moteur de Recherche mnoGoSearch 3.3.13" qui fait plus de 120 pages. Avec cette documentation, toute personne qui a un minimum de connaissance avec des outils de développement est capable de recréer, mettre à jour des composants et installer la souche mnoGoSearch. Ce travail d'industrialisation n'a pas été simple, j'ai dû récupérer l'ensemble des éléments nécessaires puis étudier comment les configurer et les intégrer au mieux dans cette souche ou produit logiciel industrialisé de mnoGoSearch. Donc, au final, l'industrialisation n'est pas un travail facile parce qu'il faut une bonne compréhension du produit au niveau de son fonctionnement, de sa configuration et de son exploitation afin ensuite d'identifier ce qui peut être industrialisé pour réduire le temps d'exécution d'une tâche pour ce produit. Lorsque l'on cherche à automatiser ou pré-configurer certains éléments, on se trouve face à l'absence d'informations pour la mise en œuvre.

Au final, je conclurai en disant que l'industrialisation peut être quelque chose d'ingrat parce que ce n'est pas créatif (bien qu'il faille parfois trouver des astuces techniques pour réaliser une tâche), de fastidieux parce qu'il ne faut rien oublier, de complexe par les connaissances et expertises nécessaires, de pointu et très large comme les connaissances des différentes polices et jeux de caractères étrangers, de coûteux en temps pour la réalisation. Mais au final, un produit industrialisé pourra être facilement mis en oeuvre et offrir un service aux utilisateurs.

VII Présentation technique du moteur de recherche retenu

Dans ce chapitre, je vais présenter la partie technique du moteur de recherche mnoGoSearch qui a été retenu suite à une évaluation.

VII.1 Architecture d'indexation

L'indexation se réalise grâce à un seul fichier binaire appelé « indexer ». Il réalise 2 traitements principaux, le premier est de récupérer en tant que navigateur web ou « spider » les documents de tout ou partie d'un site web dont l'adresse lui est fournie par les paramètres du fichier de configuration indexer.conf. Ensuite, après avoir récupéré chaque document, il les indexe suivant les directives du fichier de configuration précédemment cité et stocke ensuite les mots indexés avec différentes informations qui leur sont liées dans une ou plusieurs bases de données.

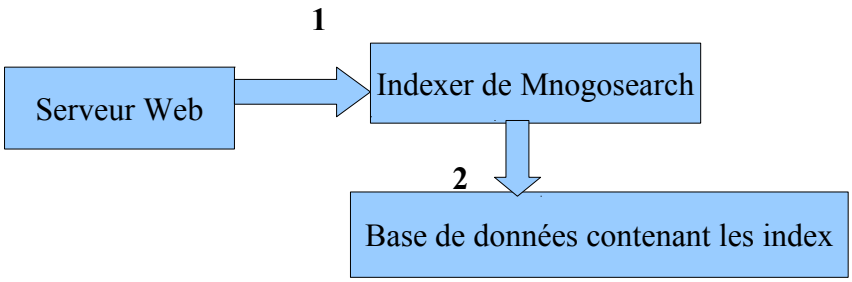

Figure 13 : Schéma des composants de l'indexation

Description des flux du schéma ci-dessus :

1: Ce flux est la récupération du texte du document à indexer qui se trouve en local ou sur un serveur distant. L'indexation se fait suivant les directives du fichier indexer.conf

2: Les mots indexés avec les informations qui leur sont liées sont stockés dans une base de données permettant les commandes SQL grâce au binaire indexer qui agit ici en tant que client SQL grâce au support des SGBD qui lui ont été fournis lors de la compilation.

VII.2 Paramétrage du moteur de recherche pour l'indexation

Comme indiqué précédemment au chapitre VI.3.3, j'ai scindé le fichier indexer.conf, presque la totalité des paramètres à transmettre au binaire indexer, en 2 parties. Le premier fichier common-indexer-latin1.conf contient les paramètres génériques pour l'indexation de documents utilisant l'encodage pour les langues d'Europe occidentale (CP1252/ISO-8859-1) et le second nommé indexer.conf charge le fichier common-indexer-latin1.conf et ensuite définit différents paramètres repris du premier fichier et décommentés. Le moteur de recherche s'appuie donc sur 2 fichiers de paramétrage qui permettent d'indexer de manière fine un site web ou une autre source de donnée quelconque.

VII.3 Les parseurs de documents bureautiques

mnoGoSearch n'intégrant que le strict minimum dans ces parseurs intégrés pour quelques types MIME de documents (text/html, text/xml, text/plain et audio/mpeg), il est nécessaire de prévoir plusieurs logiciels qui agissent en tant que parseurs externes et permettent à mnoGoSearch d'indexer d'autres types de documents. Je vais indiquer et décrire ci-dessous ces différents logiciels.

VII.3.1 Catdoc

Catdoc est un utilitaire qui lit un ou plusieurs fichiers au format binaire Microsoft Word (*.doc) et en extrait uniquement le texte vers la sortie standard, c'est à dire l'écran de l'utilisateur.

Cette sortie peut aussi être redirigée vers un fichier. Ce programme est accompagné par l'outil xls2csv qui convertit les feuilles du tableur MS Excel (*.xls) en un fichier dont les valeurs sont séparées par des virgules, et l'utilitaire catppt qui retire l'information textuelle uniquement de fichiers du logiciel de présentation MS Powerpoint (*.ppt). Ces trois programmes n'extraient que le texte avec une perte du formatage de celui-ci.

Ces outils fonctionnent suivant le même principe. D'abord, il y a la détection du numéro de jeu de caractères utilisé par le document bureautique, comme le CP1252 pour les langues de l'Europe occidentale avec MS Windows. Il est possible de forcer l'utilitaire à

utiliser un jeu de caractères précis pour traiter le document source. Ensuite on précise le jeu de caractères dans lequel sera encodé le fichier texte de destination. Il s'agit souvent d'une partie précise de la norme ISO-8859 comme la version ISO-8859-1 pour les langues d'Europe occidentale. La sortie se fera souvent en ISO-8859, puisque les utilitaires ne sont disponibles que sous le système d'exploitation Unix ou GNU-Linux et que l'ISO-8859 est la norme d'encodage standard sur ces systèmes bien que l'UTF-8 puisse aussi être utilisée.

Ces logiciels s'appuient sur des fichiers de référence définissant l'encodage de chaque lettre ou caractère fourni pour une bonne part par l'organisme ISO et disponible sur son serveur FTP.

J'ai mis à jour les utilitaires avec les dernières versions de ces fichiers de référence. J'ai aussi ajouté la détection automatique de nouveaux jeux de caractères en modifiant le code source du programme et en ajoutant les fichiers nécessaires pour effectuer l'extraction de texte.

VII.3.2 Xpdf

Xpdf est un logiciel qui a été créé pour lire les documents au format PDF d'Adobe dans une interface graphique. Ce logiciel contient un utilitaire appelé pdftotext qui réalise le même travail que ceux de Catdoc : Il lit un ou plusieurs fichiers au format binaire Portable Document (*.pdf) et en extrait uniquement le texte vers la sortie standard,

c'est-à- dire l'écran de l'utilisateur.

Cette sortie peut aussi être redirigée vers un fichier. Pour le bon fonctionnement de pdftotext, il est souhaitable de lui indiquer le jeu de caractères dans lequel la plupart des document sont encodés.

Lors de la lecture du document par pdftotext, il est possible de lui demander d'extraire les méta-données du document PDF comme le nom de l'auteur et la date. Très souvent malheureusement, ces données très utiles ne sont pas renseignées par les auteurs des documents.

VII.3.3 Parseurs de documents bureautiques utilisant XML

Le parseur de document catdoc permet de parser les principaux types de documents bureautiques de la suite Microsoft Office qui sont dans un format complètement binaire. Or depuis la version 2003 de MS Word, Microsoft a commencé à utiliser le format XML afin de stocker le texte du document bureautique et les instructions de mises en page. Face à l'utilisation importante des documents bureautiques de type MS Office en entreprise, il a été nécessaire de trouver une solution pour indexer ce type de document. Comme j'avais déjà trouvé sur Internet un script de parseur appelé sxwtoplain pour le format *.sxw de StarOffice Writer, je l'ai adapté au format *.odt d'OpenOffice Writer (norme ODF) (script odttoplain) et au format *.docx de Ms Office Word 2007 (script docxtoplain). La logique de script est très simple, il utilise l'utilitaire unzip pour décompresser le document bureautique qui n'est qu'une archive au format zip, puis il utilise l'utilitaire cat pour transmettre le contenu du fichier xml avec les données textuelles du document vers la sortie Unix standard. Ensuite l'utilitaire sed, en fonction d'une expression régulière qui lui a été transmise, extrait le contenu du document envoyé sur la sortie standard sans les marqueurs XML. Il ne reste plus qu'à transmettre ce flux de texte à mnoGoSearch pour indexation.

VII.3.4 Libpwd

C’est une librairie logicielle libre qui est utilisée pour traiter les documents WordPerfect Windows (*.wpd) et est utilisée par les logiciels bureautiques comme LibreOffice pour visualiser ou importer ce type de document. Ce logiciel contient deux utilitaires qui permmettent d’extraire le texte du document sous forme de texte brut ou au format HTML vers la sortie standard, c'est à dire l'écran de l'utilisateur.

VII.3.5 Libwps

C’est une librairie comme libpwd et fait la même chose qu’elle mais pour les documents MS Works (*.wps).

VII.3.6 SofficeToHTML

Il permet de convertir des fichiers *.sxw (StarOffice/OpenOffice 1.x Writer) au format HTML.

VII.3.7 Odftools

Il permet de convertir des documents OpenOffice/Libreoffice au format/norme ODF (Open Document Format) (*.odt) au format HTML pour les indexer ensuite par mnoGoSearch.

VII.3.8 Docx2txt

Il permet de convertir des documents Microsoft Word dans des versions 2007 et supérieures (*.docx) au format plain texte

VII.3.9 Pptx2txt

C'est un script Perl qui permet de convertir des documents Microsoft PowerPoint dans des versions 2007 et supérieures (*.pptx) au format plain texte

VII.3.10 Xlsx2csv

Il permet de convertir des documents Microsoft Excel dans des versions 2007 et supérieures (*.xlsx) au format plain texte

VII.4 Architecture de recherche

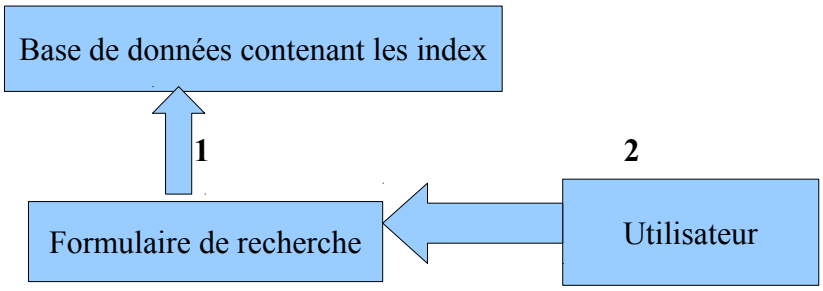

Figure 14 : Schéma des composants de la recherche

Description des flux du schéma ci-dessus :

1: Le formulaire de recherche appelle le binaire search.cgi ou module mnoGoSearch de PHP qui agissent ici en tant que clients SQL. Ce dernier récupère des résultats dans la base de données, puis les affiche à l'utilisateur au travers du fichier modèle « search.htm » qui permet de personnaliser l'affichage dans le formulaire de recherche ou via le modèle en PHP.

2 : L'utilisateur renseigne un ou plusieurs mots-clés dans le champ prévu du formulaire de recherche, puis en validant le formulaire, il envoie une requête qui fait un appel au fichier search.cgi avec les mots-clés et les paramètres annexes de la recherche. L'étape 1 ci-dessus est alors exécutée.

VII.4.1 Via module cgi

Lors de la compilation de mnoGoSearch, un binaire appelé search.cgi et fonctionnant suivant la norme CGI est créé. Ce fichier est ensuite à déposer dans le répertoire prévu pour le fonctionnement de scripts CGI dans le serveur Web HTTP comme Apache HTTPD. Après son installation, le fichier search.htm doit être paramétré dans le répertoire prévu lors de la création des binaires de mnoGoSearch. Ensuite il faut paramétrer le fichier search.htm en reprenant les mêmes paramètres (base de données, encodage, liste de mots-vides,etc...) que ceux du fichier indexer.conf ayant servi à l'indexation. Il est à noter que, s'il y a plusieurs sites différents à indexer et dont les index

sont distincts mais hébergés sur le même serveur, alors il sera nécessaire d'avoir des noms de fichiers du binaire cgi et des modèles de recherches différents en fonction de la base d'index appelée. Pour être plus explicite, si mon site s'appelle par exemple inforacle, alors le formulaire de recherche sera appelé par l'URL http://mon_serveur_de_recherche/cgi-bin/search-inforacle.cgi. Celui-ci appellera alors un modèle de recherche pour afficher les résultats dont le préfixe sera impérativement le même que celui du cgi c'est-à-dire dans notre cas : search-inforacle.htm sinon le cgi ne trouvera le bon modèle de recherche contenant les informations pour se connecter à la base des index.

Le fonctionnement en module cgi est celui qui contient le plus de fonctionnalités pour la recherche. Il en contient beaucoup plus que le fonctionnement en module PHP que nous allons maintenant décrire.

VII.4.2 Via module PHP

En plus du module cgi, les sources de mnoGoSearch contiennent le code pour PHP qui est à compiler à part si l'on souhaite ce support. Ce module contient moins de fonctionnalités et évolue moins rapidement que le module cgi. Dans le cas où l'on ne souhaite pas utiliser le module cgi pour des questions de sécurité ou si la fonction de recherche est destinée à un site web développé en PHP, alors ce module est le meilleur choix.

De plus Yannick Warnier de Beeznest a développé un script PHP qui génère une page web avec des résultats de recherche avec une sortie XML au lieu de HTML. Cela permet ensuite de parser ce flux XML afin de générer une page HTML personnalisée et de faciliter ainsi l'intégration de ce contenu dans les sites web.

VII.5 Optimisation des performances

L'optimisation des performances doit être prise en compte lors de la mise en place du moteur de recherche. Il faut bien évaluer ou déterminer le nombre de documents à indexer ou le nombre d'utilisateurs potentiels, en fonction de ces éléments il faudra faire des choix d'architectures ou de paramétrage.

- **En mode indexation**

Lors de l'indexation, il faut s'assurer que le serveur a :

- une bonne capacité au niveau de la puissance de calcul avec un processeur performant afin de réaliser rapidement les opérations de : création d'index et de traitements divers sur le texte des documents,
- une bonne carte réseau correctement configurée en fonction du débit réseau auquel le serveur est rattaché. Cette carte doit être configurée en 100Mbps full/duplex ou supérieur et non 10Mbps half/duplex si le réseau le permet,
- des disques durs rapides afin de minimiser les temps d'accès disques en lecture ou écriture et dont les paramètres systèmes sont corrects ou optimisés comme avec la commande hdparm (disque durs) ou ulimit (nombres de fichiers ouverts en simultanés) sous Linux,
- des partitions distinctes prévues pour le système, les binaires du logiciel, les données et les logs pour optimiser les accès disques et les problèmes en cas de partition pleine,
- une taille de mémoire vive correctement dimensionnée,
- une configuration du serveur de bases de données ou SGBD (Système de Gestion de Bases de Données) adaptée à la fonction du serveur (dédié ou mutualisé) et à la taille de sa mémoire (par exemple fichier pré-paramétré avec Mysql: my-medium.cnf, my-large.cnf et my-huge.cnf)
- un type de structure de base de données adaptée pour stocker les index. mnoGoSearch propose 3 modes :
    - ✗ single : les index et mots indexés sont stockés dans une table. Il peut être utilisé jusqu'à 5000 documents. Il supporte la mise à jour en direct c'est-à-dire que le « crawling » (récupération des documents et extraction des mots) et l'indexation sont réalisés en même temps.
    - ✗ multi : ce mode reprend le même principe que celui du single mais l'information sur les mots indexés est distribuée dans 256 tables distinctes en utilisant une fonction de

hachage pour la distribution. Il est possible de stocker jusqu'à 50000 documents avec cette configuration.

- ✗ blob : avec ce choix, on a le mode le plus rapide dans mnoGoSearch à la fois pour l'indexation et la recherche. De plus il permet de stocker jusqu'à 1 000 000 à 2 000 000 millions de documents sur une seule machine. Par contre le « crawling » se fait dans un premier temps puis ensuite la création des index. Ces 2 étapes nécessitent de lancer le programme « indexer » en 2 temps avec une ligne de commande différente comme vu précédemment.

- suivant le nombre de processeurs et leur puissance, il sera possible de lancer plusieurs processus du même programme « indexer » en parallèle afin d'accélérer le temps de traitements des sources de données à indexer. Pour cela, on utilise le paramètre -N suivi du nombre de threads et qui est passé au programme « indexer » lors de son lancement.

- **En mode recherche**

  - Lors de la recherche, en plus des points cités plus haut pour l'indexation, il faut s'assurer que le serveur a :
  - un serveur Web HTTP correctement configuré et avec un nombre de processus importants afin de servir les requêtes des utilisateurs en cas de fortes charges,
  - une utilisation d'une liste de mots-vides qui a dû être activée lors de l'indexation afin de réduire la taille de la base de données et de permettre des recherches rapides,
  - une utilisation des synonymes et racines des mots pour avoir plus de résultats pertinents lors des recherches via Ispell qui soit adaptée et activée si cela est réellement nécessaire. L'utilisation d'Ispell avec mnoGoSearch est déconseillée parce qu'elle entraîne une forte charge du serveur en cas de recherches.

VII.6 Intégration du moteur dans les sites web

Lorsque l'on parle de l'intégration, il s'agit d'intégrer la page contenant le champ du ou des mots à rechercher et la page d'affichage des résultats de la recherche. Pour le cgi, il sera nécessaire de créer une page de recherche avec un formulaire dont un des champs servira à transmettre le ou les mots recherchés via une requête HTTP POST au binaire search.cgi qui affichera alors les résultats de la recherche via le modèle de recherche qui lui est associé. Ce modèle en HTML possède une mise en page particulière par défaut. Dans le cadre de mon mémoire, il m'a été nécessaire de créer une page web de recherche et d'intégrer le modèle d'affichage des résultats de recherche avec les chartes graphiques web d'EDF et GDF pour leur Intranet. Cette intégration s'est faite sans difficulté grâce à l'utilisation de fichier de style CSS et de scripts en Javascript. Comme ce modèle de recherche search.htm fourni avec mnoGoSearch ne répond pas toujours au besoin des commanditaires de sites web, il est possible de le modifier en retirant par exemple la fonction de recherche avancée pour ne garder que la fonction de recherche simple. Pour le PHP, il suffit juste d'appeler la page index.php qui sert à la fois de page de formulaire de recherche et d'affichage de recherche. En cas de charte graphique, il faut modifier ce fichier index.php et intégrer la charte et faire les autres modifications de mise en page.

Formulaire de recherche avec charte Intranet EDF :

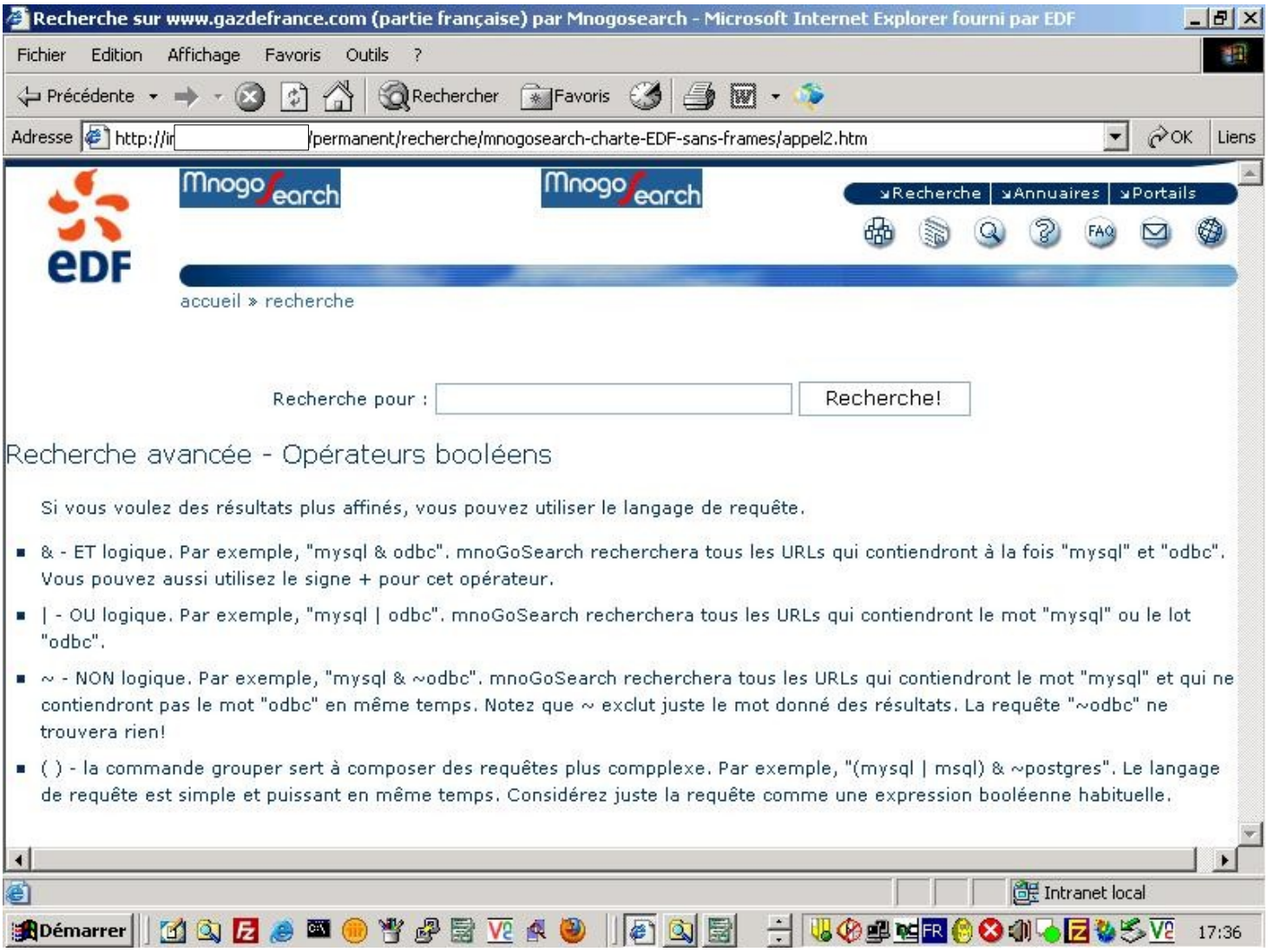

Figure 15 : Capture d'écran du formulaire de recherche avec charte Intranet EDF

Formulaire de résultats avec charte Intranet EDF :

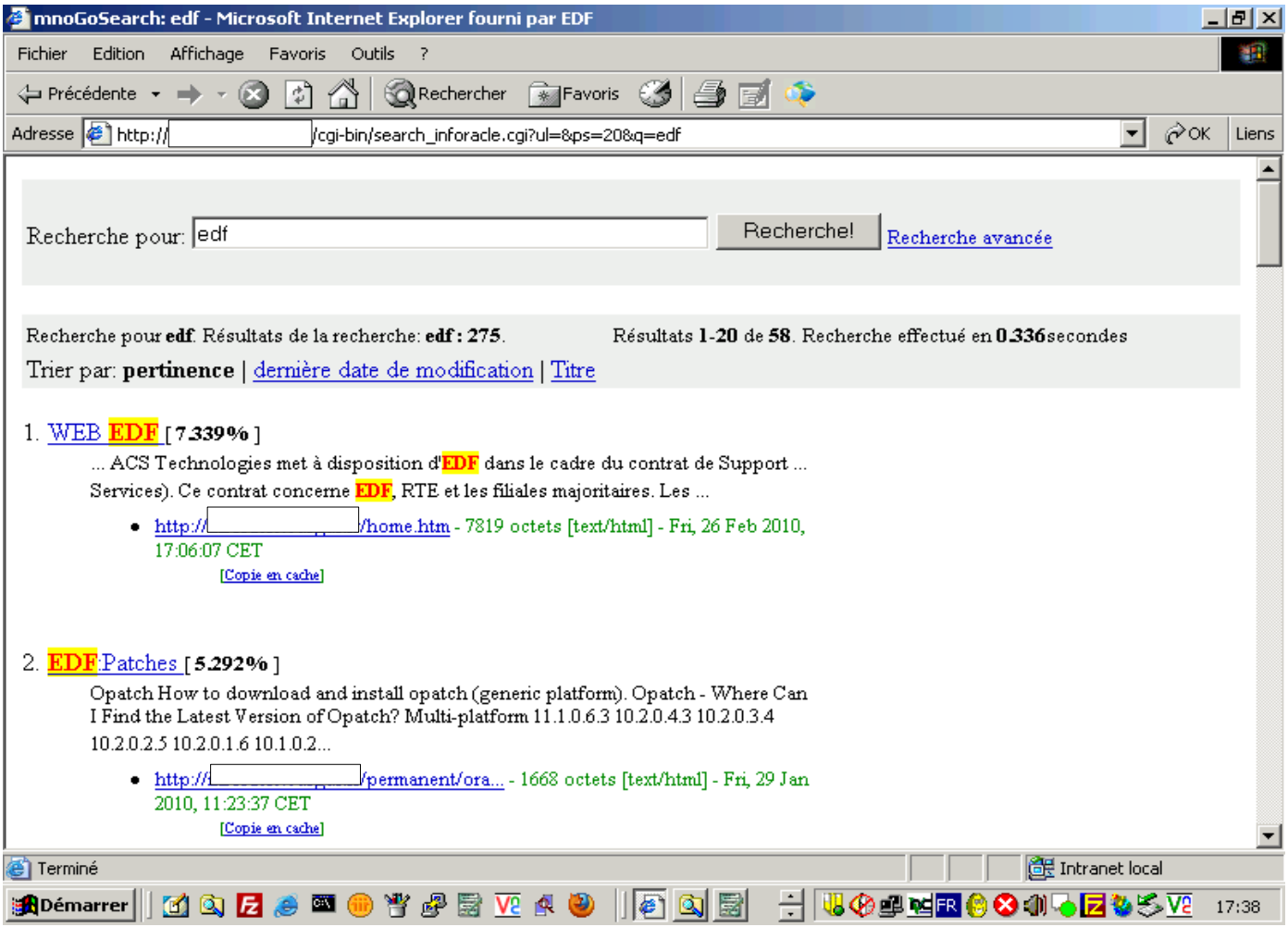

Figure 16 : Capture d'écran du formulaire de résultats avec charte Intranet EDF

Formulaire de recherche avec charte Intranet GDF :

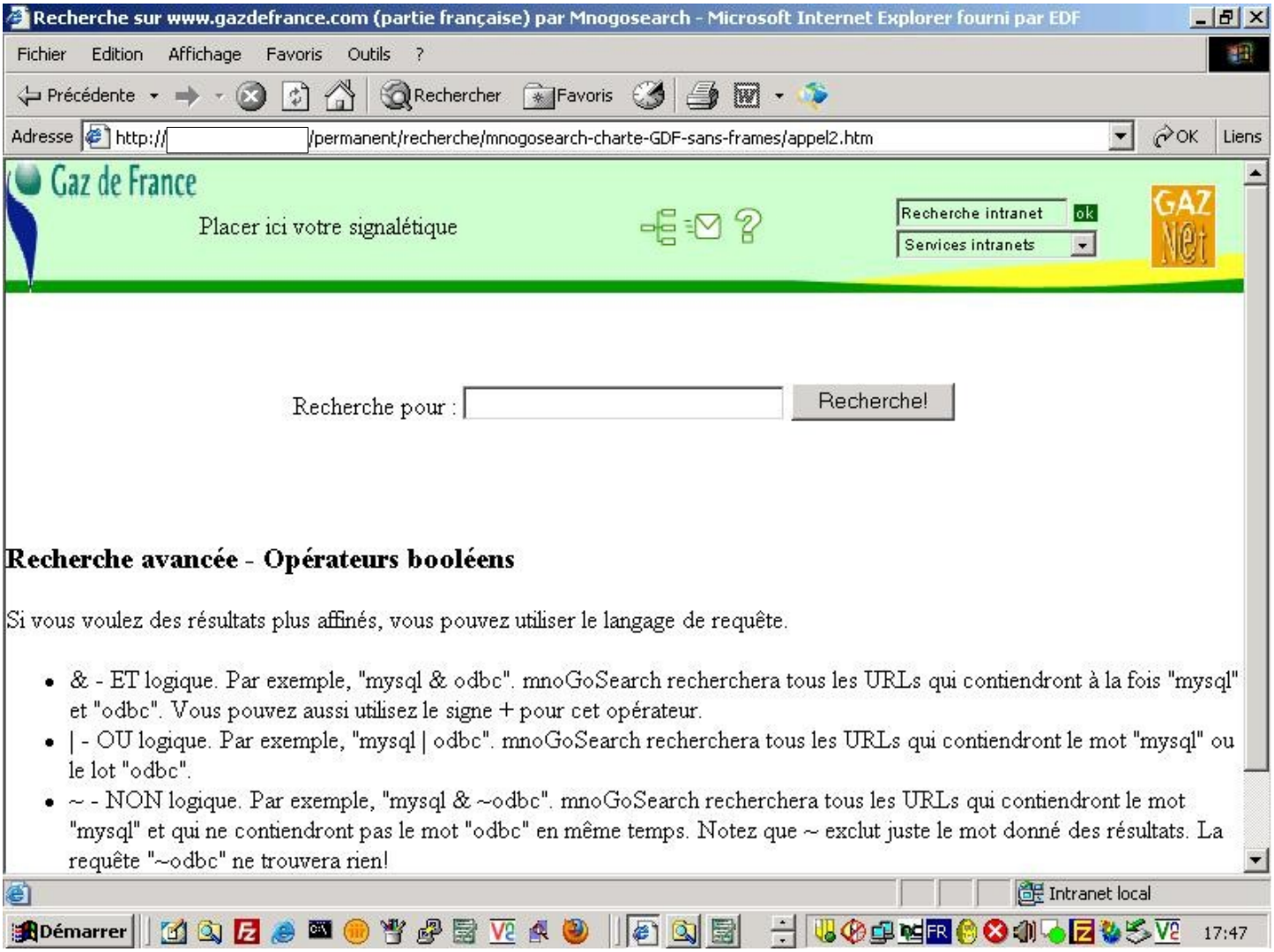

Figure 17 : Capture d'écran du formulaire de recherche avec charte Intranet GDF

Formulaire de résultats avec charte Intranet GDF :

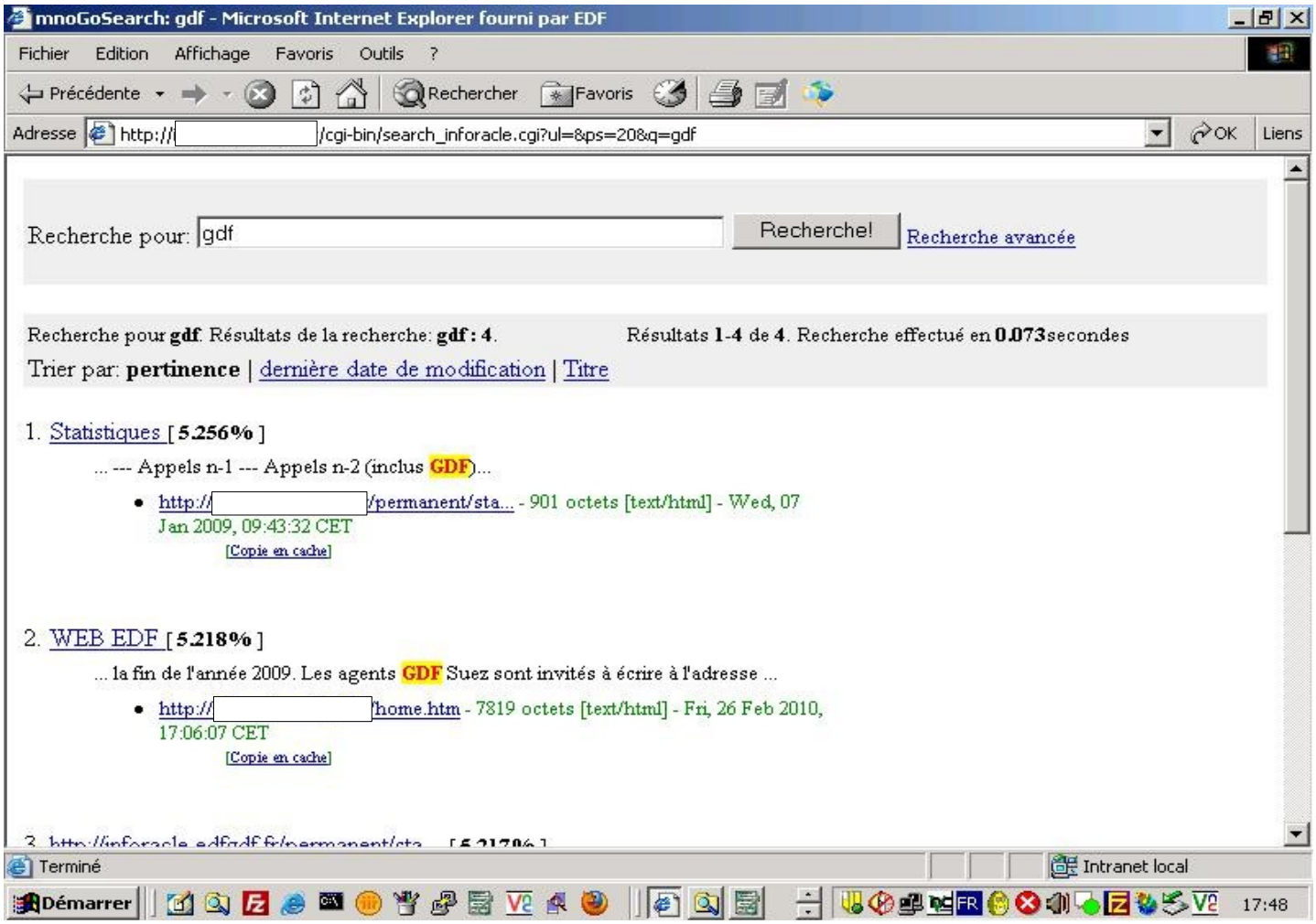

Figure 18 : Capture d'écran du formulaire de résultats avec charte Intranet GDF

VII.7 La sécurité au sein de mnoGoSearch

mnoGoSearch est prévu pour accéder à des sites sécurisés. Il supporte le protocole HTTPS et permet donc d'indexer des sites web sécurisés via un certificat SSL. Certains sites web sont aussi sécurisés par une authentification par identifiant et mot de passe qui sont demandés par le serveur HTTP. Dans ce cas mnoGoSearch est capable de renvoyer la paire identifiant/mot de passe plus le le nom attendu pour le domaine afin de passer la barrière et d'effectuer l'indexation. Des sites réalisés en PHP nécessitent de transmettre un identifiant et un mot de passe dans un formulaire afin de créer un jeton de connexion qui se traduira par la création d'une session PHP dont l'identifiant sera stocké dans un fichier cookies par le navigateur web sur le poste client. Il suffit de récupérer le contenu de ce fichier « cookies » et de l'indiquer dans le fichier indexer.conf afin de réaliser l'indexation d'un site en PHP. mnoGoSearch contient une fonction de cache qui permet d'afficher le texte du document avec le mot recherché en surbrillance (voir captures d'écrans précédemment). Ainsi, afin d'éviter une fuite d'informations en cas de récupération de l'URL de recherche, il vaut mieux désactiver cette fonction de cache dans le fichier indexer.conf. Cette précaution peut être nécessaire pour les sites avec des données sensibles. L'accès au document à partir d'un résultat de recherche nécessitera alors l'authentification de l'utilisateur vis-à vis du site sécurisé.

VIII Conclusion

VIII.1 Récapitulatif du travail réalisé

Mon travail dans le cadre de ce mémoire d'ingénieur a consisté à réaliser en premier un cahier des charges avec l'ensemble des fonctionnalités obligatoires et optionnelles nécessaires au remplacement du moteur de recherche Verity Search'97 existant. Ensuite j'ai fait une liste assez exhaustive de plusieurs dizaines de moteur de recherche en logiciels libres existants sur Internet et j'en ai éliminé une bonne partie sur 2 principaux critères qui sont le support du protocole HTTPS et d'avoir eu au moins une mise à jour dans les 12 derniers mois afin d'écarter les projets morts. Après j'ai retenu la méthode d'évaluation de logiciels libres appelée QSOS. En l'absence de grille d'évaluation disponible pour cette dernière, j'en ai créé une avec différents critères en me basant sur les différents documents disponibles sur Internet qui comparaient différents moteurs de recherche entre eux. Après cette étape, j'ai recherché des informations détaillées sur chacun des moteurs et j'ai fait l'évaluation objective de ceux-ci via la grille d'évaluation QSOS. Le moteur de recherche retenu a été mnoGoSearch parce qu'il correspondait le mieux aux besoins d'EDF pour le remplacement de l'ancien moteur et permettait d'évoluer grâce à des parseurs externes pour indexer de nouveaux types de documents. J'ai ensuite industrialisé ce logiciel en essayant de l'étendre en fonctionnalités pour l'indexation et en lui adjoignant différents parseurs de documents bureautiques, en pré-paramétrant au maximum le fichier de configuration de l'indexation, en améliorant et en localisant le formulaire de recherche en français, en scriptant les principales tâches (indexation, création et suppression des bases de données de stockage des index, affichage des statistiques d'indexation). L'industrialisation a nécessité, en plus du cahier des charges, la rédaction de plusieurs documents: document de constitution du logiciel industrialisé, procédures d'installation et d'exploitation, archive des binaires et des fichiers de configuration prêts à être installés sur le serveur.

VIII.2 Les déploiements passés du moteur

Le moteur de recherche mnoGoSearch a été utilisé avec succès sur 2 sites par le passé. Le premier est le site PSG (Portail Salarié Groupe) entre le milieu des années 2000 et le début des années 2010. Ce site n'existe plus à l'heure actuelle. Sur ce site web, le moteur de recherche mnoGoSearch a rempli avec succès sa tache en permettant aux utilisateurs de faire des recherches dans un site multilingue en français, anglais et allemand. Pour les collections des index en allemand et en français, la maîtrise d'ouvrage m'a fait remonter un problème avec la gestion des caractères accentués, l'utilisateur obtenait des résultats à sa recherche uniquement avec le mot-clé renseigné et non avec ce même mot-clé qu'il soit avec ou sans accents. J'ai donc du trouver une solution pour résoudre ce problème en utilisant la collation allemande dans Mysql. Au final le système a bien fonctionné et par exemple, si l'utilisateur tapait le mot clé « électricité » sans accent, il avait quand même des résultats à sa recherche contenant ce mot-clé avec ou sans accent.

Le deuxième est le site inforacle qui contient beaucoup de documentations sur le SGBD Oracle DB et dont le contenu est mis à disposition des utilisateurs sur un site web dédié. Le moteur de recherche mnoGoSearch a permis de faciliter les recherches et de permettre aux utilisateurs de trouver plus rapidement des informations dans ce gros ensemble de documents.

VIII.3 Les déploiements actuels possibles du moteur

Comme vu précédemment, le moteur de recherche retenu permet d'indexer différents types de documents accessibles via différents protocoles réseau. Actuellement un service d'indexation et de recherche à base du serveur autonomy Idol est disponible sur l'intranet et l'internet d'EDF. mnoGoSearch pourrait donc répondre à des besoins spécialisés sur des sites web EDF avec de l'information sensible sur l'Internet dont l'accès est restreint par comptes nominatifs et qui nécessitent une fonctionnalité de recherche interne puisque des moteurs de recherche comme Google ne fonctionne pas dans ce cas. mnoGoSearch pourrait aussi servir comme moteur de recherche d'appoint pour indexer des documents dont le support n'est pas pris en charge par le moteur d'entreprise EDF qui est Autonomy Idol V7 mais dont une extraction du contenu textuel pourrait être faite par un parseur externe, support ce type de format comme celui de compression *.xz de plus en

plus répandu sur Linux. mnoGoSearch a la possibilité de répondre à des besoins d'indexation de langues dont le support n'a pas été acheté par EDF pour Autonomy Idol.

VIII.4 Les coûts

Le coût global pour EDF pourrait comprendre le coût d'un support chez la société qui développe le logiciel pour un montant de 950 Dollars US par an, en mai 2013, avec un support de base par courriel ou de 6500 Dollars US par an avec un support étendu qui comprendra un paramétrage personnalisé de l'installation mnoGoSearch du client et des développements spécifiques qui seront inclus dans la prochaine version de mnoGoSearch. A ce coût initial, il faudrait rajouter une partie du coût en salaire de la ou les personnes qui seraient chargées de faire le support sur mnoGoSearch. Dans ce cas, on peut les mutualiser avec l'équipe déjà en charge du support sur Autonomy Idol. Concernant l'industrialisation qui nécessite des compétences de compilation de code source logiciel, elle pourrait être faite par le service CC_WEX/ESA pour environ une charge de 120 à 150 jours qui se reproduirait tous les 2 à 3 ans.

VIII.5 Bilan personnel

Ce logiciel n'a pas été facile à industrialiser parce qu'il demande beaucoup de compétences différentes sur différentes briques logicielles comme les serveurs web, les SGBD, les formats de documents bureautiques, la sécurité des sites web, la compilation, langage HTML et les scripts Shell ou Perl. Il faut donc intégrer l'ensemble de ces connaissances pour faire une bonne industrialisation d'un logiciel libre de moteur de recherche.

VIII.6 L'avenir des moteurs de recherche d'entreprise

Il y aura toujours besoin de moteur de recherche d'entreprise sur les sites web des entreprises pour 2 raisons principales : ces sites web privés ne peuvent pas être indexés par un moteur de recherche public comme Google, Yahoo ou Bing, et je vois mal laisser un site web Intranet même bien conçu au niveau des menus avec beaucoup de documentations

ou d'informations sans moteur de recherche intégré parce que l'utilisateur passerait beaucoup de temps à trouver ce qu'il cherche et finirait par se lasser.

L'avenir est peut-être dans la puissance du web sémantique avec un outil comme KnowledgeGraph chez Google [KnowledgeGraph] qui est une des structures centrales sous-jacentes de ces services et qui permet d'envisager des analogies entre différents types de documents, mais aussi en savoir plus sur vous et vous profiler pour optimiser les annonces publicitaires. Par contre ce type de technologie nécessite d'agréger et de traiter des informations de différentes sources et des documents bureautiques mais aussi de type multimédia. Elle ne peut donc qu'être réservée à de grosses organisations ou entreprises par ces coûts. Coté entreprise, cela pourrait permettre d'identifier des comportements suspects comme de l'espionnage industriel ou permettre à des chercheurs de faire des corrélations non prévues dans le cadre de recherches scientifiques. C'est actuellement le cas avec le traitement de grosses données textuelles d'études médicales par des statisticiens du « Big Data » pour trouver de nouveaux axes de traitements des maladies [BIGDATA-MEDICAL].

Bibliographie

Moteurs de recherche d'entreprise open source.

[WEB-ST] Avi Rappoport. Search Tools for Web Sites and Enterprise, [en ligne]. Disponible sur : <http://www.searchtools.com>. (consulté le 18/05/2011). Site sur les outils de recherche pour les sites web et intranets.

[WEB-MNOGO] Alexander Barkov. mnoGoSearch web search engine software, [en ligne]. Disponible sur : <http://www.mnogosearch.org>. (consulté le 18/05/2011). Site du moteur de recherche mnoGoSearch.

[WEB-HTDIG] The ht://Dig Group. WWW Search Engine Software , [en ligne]. Disponible sur : <http://www.htdig.org/>. (consulté le 18/05/2011). Site du moteur de recherche ht://Dig.

[WEB-NUTCH] The Apache Software Foundation. Welcome to Nutch!, [en ligne]. Disponible sur : <http://nutch.apache.org/>. (consulté le 18/05/2011). Site du moteur de recherche Nutch.

[WEB-BN] ht://Dig vs mnoGoSearch Comparison. In : Yannick Warnier. BeezNest Open-Source specialists, [en ligne]. Disponible sur : <http://beeznest.wordpress.com/2005/02/14/htdig-vs-mnogosearch-comparison/>. (consulté le 18/05/2011). Comparaison de ht://Dig et mnoGoSearch.

[WEB-ACACIA] Cours. In : Groupe ACACIA de l'INRIA. Sophia Antipolis - Méditerranée - Centre de recherche INRIA - Institut national de recherche en informatique et automatique, [en ligne]. Disponible sur : <http://www-sop.inria.fr/acacia/cours/>. (consulté le 18/05/2011). Site sur le web sémantique.

[WEB-WRG] A Comparison of Open Source Search Engines. In : Ricardo BAEZA-YATES, Christian MIDDLETON. Web Research Group, [en ligne]. Disponible sur : <http://wrg.upf.edu/WRG/dctos/Middleton-Baeza.pdf>. Octobre 2007. (consulté le

18/05/2011). Comparaison de 29 moteurs de recherche sur de très gros volumes de données de750Mo à 10,2Go en HTML.

[WEB-JDN] Enrichir son intranet avec des briques Open Source en CMS, GED et moteur de recherche. In : Dominique FILIPPONE. Le Journal du Net : e-Business, Informatique, Economie et Management, [en ligne]. Disponible sur : <http://www.journaldunet.com/solutions/intranet-extranet/enquete/08/0325-open-source-intranet/7.shtml
>. (consulté le 18/05/2011).

[WEB-CNR.IT] BUZZI Marina LAZZARESCHI Pasquale. A comparison between public-domain search engines. In : Conferenza Annuale CMG Italia, Rome - Italie , 8-10 Mai 2006. , [en ligne]. Disponible sur : <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=FCDB7A5A578D43688DAE403FF81C2643?doi=10.1.1.89.9661&rep=rep1&type=pdf>. (consulté le 18/05/2011). Les auteurs de l'Italian National Research Council comparent Nutch, mnoGoSearch, DataparkSearch et ht//Dig. mnoGoSearch arrive en tête pour la montée en charge, la pertinence des résultats, la robustesse et la rapidité de récupération et d'indexation des documents.

Moteurs de recherche sur Internet.

[WEB-GOOGLE] Google. Google France, [en ligne]. Disponible sur : <http://www.google.fr/>. (consulté le 18/05/2011). Le site web Google.

[WEB-YAHOO] Yahoo. Yahoo! France, [en ligne]. Disponible sur : <http://fr.yahoo.com/>. (consulté le 18/05/2011). Le site web Yahoo.

[WEB-VOILA] Voila. Voila, actualité, e-mail, t'chat, traduction, RSS, web 2.0 et moteur de recherche, [en ligne]. Disponible sur : <http://www.voila.fr>. (consulté le 18/05/2011). Le site web Voila.

[WEB-S2M] 1ère Position SA. S2M : Actualités et veille référencement SEO SEA SMO : Google, Facebook, Twitter, Bing, [en ligne]. Disponible sur : <http://www.secrets2moteurs.com>. (consulté le 18/05/2011). Indice audience moteurs de recherche Internet.

[WEB-SEW] Incisive Interactive Marketing LLC.. Search Engine Marketing (SEM), Paid Search Advertising (PPC) & Search Engine Optimization (SEO) - Search Engine Watch (#SEW), [en ligne]. Disponible sur : <http://searchenginewatch.com/>. (consulté le 18/05/2011). Veille sur les moteur de recherche Web (en anglais).

[WEB-ABOND] Olivier Andrieu. Abondance : l'actualité et l'information sur le référencement et les moteurs de recherche, [en ligne]. Disponible sur : <http://www.abondance.com/>. (consulté le 18/05/2011). Veille sur les moteur de recherche Web (en français).

[WEB-MOTRECH] Jérôme Charron. motrech, [en ligne]. Disponible sur : <http://motrech.free.fr/>. (consulté le 18/05/2011). Veille sur les moteur de recherche Web (en français).

[WEB2010] Maurice de Kunder.The size of the World Wide Web, [en ligne]. Disponible sur : <http://www.worldwidewebsize.com/>. (consulté le 18/05/2011).

[DEEP-WEB] White Paper: The Deep Web: Surfacing Hidden Value. In : Michael K. Bergman. MLibrary Digital Collections, [en ligne]. Disponible sur : <http://quod.lib.umich.edu/cgi/t/text/text-idx?
c=jep;view=text;rgn=main;idno=3336451.0007.104>. (consulté le 18/05/2011).

[WEB2002] Internet Statistics: Distribution of languages on the Internet. In : Inconnu. Netz-Tipp.De - Buchhandel, Verlag, Schule, Literatur - Startseite, [en ligne]. Disponible sur : <http://www.netz-tipp.de/languages.html>. (consulté le 18/05/2011).

[WEB2005] The Indexable Web is more than 11.5 billion pages. In : Antonio Gulli - Alessio Signorini. Department of Computer Science - The University of Iowa, [en ligne]. Disponible sur : <http://www.cs.uiowa.edu/~asignori/papers/the-indexable-web-is-more-than-11.5-billion-pages/>. (consulté le 18/05/2011).

[WEB2008] We knew the web was big.... In : Jesse Alpert & Nissan Hajaj. Official Google Blog, [en ligne]. Disponible sur : <http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html>. (consulté le 18/05/2011).

[WEB2009] INTERNET WORLD USERS BY LANGUAGE - Top 10 Languages. In : Miniwatts Marketing Group.. Internet World Stats - Usage and Population Statistics, [en ligne]. Disponible sur : <http://www.internetworldstats.com/stats7.htm>. (consulté le 18/05/2011).

[DOMAIN2010] Domain Counts & Internet Statistics. In : DomainTools, LLC . DomainTools | Whois Lookup, DNS Lookup, Reverse Whois Lookup, [en ligne]. Disponible sur : <http://www.domaintools.com/internet-statistics/>. (consulté le 18/05/2011).

[MDR98] Statistiques sur les moteurs de recherche. In : InternetDiffusion. Guide complet des Moteurs de Recherche, [en ligne]. Disponible sur : <http://www.lesmoteursderecherche.com/stats.htm>. (consulté le 18/05/2011).

[MDR04] Etude Nielsen Netratings - Juillet 2004. In : Nielsen Netratings. indicateur - moteur de recherche, site d'information référencement & positionnement, [en ligne]. Disponible sur : <http://www.indicateur.com/barometre/0407-etude-nielsen-internat.asp>. (consulté le 18/05/2011).

[MDR10] Nielsen Provides Topline U.S. Web Data for March 2010. In : Nielsen Netratings. Nielsen Wire, [en ligne]. Disponible sur : <http://blog.nielsen.com/nielsenwire/online_mobile/nielsen-provides-topline-u-s-web-data-for-march-2010/>. (consulté le 18/05/2011).

[TEMPS] France : Les usages du Web par les Internautes. In : CCM Benchmark. Le Journal du Net : e-Business, Informatique, Economie et Management, [en ligne]. Disponible sur : <http://www.journaldunet.com/cc/01_internautes/inter_usage_fr.shtml>. (consulté le 18/05/2011).

[MDRSTAT] Research and Statistics. In : Web Marketing Workshop Ltd. Search engine optimisation: Web Search Workshop, UK SEO, PPC and website optimisation specialists., [en ligne]. Disponible sur : <http://www.websearchworkshop.co.uk/stats.php>. (consulté le 18/05/2011).

[iprospect] iProspect Search Engine User Attitudes. In : iProspect. Search Engine Marketing Firm for SEO - PPC Services | iProspect, [en ligne]. Disponible sur : <http://www.iprospect.co.th/about/whitepaper_surveycomplete.htm ou http://www.iprospect.co.th/premiumPDFs/iProspectSurveyComplete.pdf >. (consulté le 18/05/2011).

[KnowledgeGraph] La nouvelle arme absolue de Google n'était-elle pas plutôt son Knowledge Graph ?. In : iProspect. Usine Digitale > Blog, [en ligne]. Disponible sur : <http://www.usinenouvelle.com/article/la-nouvelle-arme-absolue-de-google-n-etait-elle-pas-plutot-son-knowledge-graph.N197114 >. (consulté le 10/06/2013).

[BIGDATA-MEDICAL] Building tools to promote sound health decisions.. In : iProspect. Symcat Blog, [en ligne]. Disponible sur : <http://blog.symcat.com/post/34831670375/how-the-big-data-trend-will-support-medical-research >. (consulté le 10/06/2013).

Informations sur la conception et fonctionnement d'un moteur de recherche

[WEB-MOONEY] CS 371R: Information Retrieval and Web Search. In : Raymond J. Mooney. Computer Science Department | The University of Texas at Austin, [en ligne]. Disponible sur : <http://www.cs.utexas.edu/~mooney/ir-course/>. (consulté le 18/05/2011).Cours sur la recherche d'information et la recherche Web

[MDR-IR] Introduction to Information Retrieval. In : Christopher D. Manning, Prabhakar Raghavan et Hinrich Schütze. The Stanford NLP (Natural Language Processing) Group, [en ligne]. Disponible sur : <http://nlp.stanford.edu/IR-book/information-retrieval-book.html>. (consulté le 18/05/2011).

[MDR-UQAM1] INF 6104 - Recherche d'informations et web. In : Daniel Lemire. Télé-université, Université du Québec à Montréal, [en ligne]. Disponible sur : <http://benhur.teluq.uqam.ca/SPIP/inf6104/>. (consulté le 13/06/2013).

[MDR-UQAM2] INF 6460 - Recherche et filtrage d'informations. In : Daniel Lemire. Télé-université, Université du Québec à Montréal, [en ligne]. Disponible sur : <http://benhur.teluq.uqam.ca/SPIP/inf6460/>. (consulté le 13/06/2013).

Table des annexes[1]

[1] Les annexes doivent être annoncées dans le texte principal en note de bas de page. On évitera alors de renvoyer à la page où se situe l'annexe mais on renverra plutôt au n° de l'annexe. On peut ici détailler ou illustrer des informations qui n'ont pas pu être développées dans le texte mais qui méritent de l'être. Les annexes sont numérotées et titrées. On évitera donc de faire figurer plusieurs annexes sur une même page. Pour enlever cette note de bas de page, supprimer l'appel de note ci-dessus.

Annexe 1
Extraits du fichier common-indexer-latin1.conf

Extraits du fichier avec les parties en gras indiquant le texte ajouté ou modifié par rapport à la configuration du fichier d'origine indexer.conf-dist de mnoGoSearch 3.3.14 :

```
##############################################################################
# Fichier de configuration de MnogoSearch pour les langues d'Europe occidentale
##############################################################################

#!/usr/local/mnogosearch/sbin/indexer -d

...

##############################################################################
#  Section 1.
#  Global parameters.

##############################################################################
# DBAddr <URL-style database description>
# Options (type, host, database name, port, user and password)
# to connect to SQL database.
# Should be used before any other commands.
# Has global effect for whole config file.
# Format:
#DBAddr <DBType>:[//[DBUser[:DBPass]@]DBHost[:DBPort]]/DBName/[?dbmode=mode]
#
# ODBC notes:
#                 Use DBName to specify ODBC data source name (DSN)
#                 DBHost does not matter, use "localhost".
#
# Currently supported DBType values are
# mysql, pgsql, mssql, oracle, ibase, db2,
# mimer, monetdb, sqlite, sqlite3, sybase, virtuoso.
#
# MySQL users can specify path to Unix socket when connecting to localhost:
#DBAddr mysql://foo:bar@localhost/mnogosearch/?socket=/tmp/mysql.sock
#
# If you are using PostgreSQL and do not specify the hostname part:
#DBAddr       pgsql://user:password@/dbname/
# then indexer will connect through the Unix socket instead of a TCP port.
#
# SQLite3 example (the trailing slash is required):
#DBAddr sqlite3:///$(VarDir)/mnogosearch.sqlite3/
#
```

 # You may also select database mode of word storage.

 # When "single" is specified, all words are stored in the same table.

 # If "multi" is selected, words will be located in different tables.

 # "multi" mode is usually faster but requires more tables.

 # Default mode is "single".

 # Parametrer la base pour stocker les index ici

 # syntaxe DBAddr mysql://login:mot_de_passe@IP_ou_nom_hote/nom_de_la_base/?dbmode=blob&ps=yes

 # Le parametre ps=yes necessite mnogosearch >= 3.3.8 ET Mysql >= 4.1.x

#DBAddr mysql://root@localhost/test/?dbmode=blob

###

VarDir /usr/local/mnogosearch/var

You may choose alternative working directory for

search results cache:

#

#VarDir /usr/local/mnogosearch/var

#VarDir /var/mnogosearch_3.3.14/cache

. . .

###

StopwordFile <filename>

Load stop words from the given text file. You may specify either absolute

file name or a name relative to mnoGoSearch /etc directory. You may use

several StopwordFile commands.

#

#StopwordFile stopwords/en.sl

#Include stopwords.conf

#fichier des mots vides francais (permet de reduire la taille des index utiles)

#StopwordFile stopwords/fr.sl

#fichier des mots vides anglais

#StopwordFile stopwords/en.sl

###

LangMapFile <filename>

Load language map for charset and language guesser from the given file.

You may specify either an absolute file name or a name relative

to mnoGoSearch /etc directory. You may use several LangMapFile commands.

#

#LangMapFile langmap/en.ascii.lm

#Include langmap.conf

 # Detection langue francaise

#LangMapFile langmap/fr.latin1.lm

#LangMapFile langmap/fr.latin1.bible.lm

 # Detection langue anglaise

#LangMapFile langmap/en.ascii.lm

. . .

##

MaxDocSize bytes

Default value 1048576 (1 Mb)

Takes global effect for whole config file

#MaxDocSize 1048576

 # 50 Mb ou Mo

MaxDocSize 52428800

. . .

##

WordCacheSize bytes

Default value 8388608 (8 Mb)

Defines maximal in-memory words cache size.

Note: cache is allocated for every DBAddr, so if you have 3 DBAddr

commands and WordCacheSize is 10Mb, it can take up to 30Mb of memory.

#WordCacheSize 8388608

 # 50 Mb ou Mo

WordCacheSize 52428800

. . .

##

#Disallow [Match|NoMatch] [NoCase|Case] [String|Regex] <arg> [<arg> ...]

Use this to disallow URLs that match (doesn't match) given argument.

The meaning of first three optional parameters is exactly the same

with "Allow" command.

You can use several arguments for one 'Disallow' command.

Takes global effect for config file.

#

Examples:

Disallow URLs that are not in udm.net domains using "string" match:

#Disallow NoMatch *.udm.net/*

Disallow any except known extensions and directory index using "regex" match:

#Disallow NoMatch Regex \/$|\.htm$|\.html$|\.shtml$|\.phtml$|\.php$|\.txt$

Exclude cgi-bin and non-parsed-headers using "string" match:

#Disallow */cgi-bin/* *.cgi */nph-*

Exclude anything with '?' sign in URL. Note that '?' sign has a

special meaning in "string" match, so we have to use "regex" match here:

#Disallow Regex \?

Disallow document extensions that are not understood by default.

Comment these lines if you have corresponding external parsers.

#Disallow *.rtf

#Disallow *.doc

#Disallow *.xls

#Disallow *.ppt

#Disallow *.pdf

Exclude some known extensions using fast "String" match:

Disallow *.b *.sh *.md5 *.rpm

Disallow *.arj *.tar *.zip *.tgz *.gz *.z *.bz2

Disallow *.lha *.lzh *.rar *.zoo *.ha *.tar.Z

Disallow *.gif *.jpg *.jpeg *.bmp *.tiff *.tif *.xpm *.xbm *.pcx

Disallow *.vdo *.mpeg *.mpe *.mpg *.avi *.movie *.mov *.wmv

Disallow *.mid *.mp3 *.rm *.ram *.wav *.aiff *.ra

Disallow *.vrml *.wrl *.png *.ico *.psd *.dat

Disallow *.exe *.com *.cab *.dll *.bin *.class *.ex_

Disallow *.tex *.texi *.texinfo

Disallow *.cdf *.ps

Disallow *.ai *.eps *.hqx

Disallow *.cpt *.bms *.oda *.tcl

Disallow *.o *.a *.la *.so

Disallow *.pat *.pm *.m4 *.am *.css

Disallow *.map *.aif *.sit *.sea

Disallow *.m3u *.qt

Disallow *.swf *.js

Exclude Apache directory list in different sort order using "string" match:

Disallow *D=A *D=D *M=A *M=D *N=A *N=D *S=A *S=D

More complicated case. RAR .r00-.r99, ARJ a00-a99 files

and UNIX shared libraries. We use "Regex" match type here:

Disallow Regex \.r[0-9][0-9]$ \.a[0-9][0-9]$ \.so\.[0-9]$

. . .

###

#AddType [String|Regex] [Case|NoCase] <mime type> <arg> [<arg>...]

This command associates filename extensions (for services

that don't automatically include them) with their mime types.

Currently "file:" protocol uses these commands.

Use optional first two parameter to choose comparison type.

Default type is "String" "NoCase" (case sensitive string match with

'?' and '*' wildcards for one and several characters correspondingly).

```
#
AddType image/x-xpixmap    *.xpm
AddType image/x-xbitmap    *.xbm
AddType image/gif          *.gif

AddType      text/plain                          *.txt  *.pl *.js *.h *.c *.pm *.e
AddType      text/html                           *.html *.htm
AddType      text/xml                            *.xml
AddType      message/rfc822                      *.eml *.mht *.mhtml

AddType text/rtf                                 *.rtf
AddType application/pdf                          *.pdf
AddType application/msword              *.doc
AddType application/vnd.ms-excel        *.xls
AddType application/vnd.ms-powerpoint   *.ppt
AddType text/x-postscript               *.ps

AddType application/vnd.sun.xml.writer    *.sxw

AddType application/vnd.oasis.opendocument.presentation *.odp
AddType application/vnd.oasis.opendocument.chart *.odc
AddType application/vnd.oasis.opendocument.spreadsheet *.ods
AddType application/vnd.oasis.opendocument.text *.odt

AddType application/vnd.openxmlformats-officedocument.presentationml.presentation *.pptx
AddType application/vnd.openxmlformats-officedocument.spreadsheetml.sheet *.xlsx
AddType application/vnd.openxmlformats-officedocument.wordprocessingml.document *.docx

AddType application/zip                          *.zip
AddType application/x-gzip              *.gz
AddType application/x-tar               *.tar
AddType application/x-lha               *.lha
AddType application/x-arj-compressed    *.arj
AddType application/x-rar-compressed    *.rar
AddType application/x-ace-compressed    *.ace

# You may also use quotes in mime type definition
# for example to specify charset. e.g. Russian webmasters
# often use *.htm extension for windows-1251 documents and
# *.html for UNIX koi8-r documents:
#
#AddType "text/html; charset=koi8-r"      *.html
#AddType "text/html; charset=windows-1251" *.htm
#
# More complicated example for rar .r00-r.99 using "Regex" match:
#AddType Regex application/rar  \.r[0-9][0-9]$
#
# Default unknown type for other extensions:
AddType      application/unknown *.*
```

```
# Mime <from_mime> <to_mime> <command line>
#
# This is used to add support for parsing documents with mime types other
# than text/plain and text/html. It can be done via external parser (which
# must provide output in plain or html text) or just by substituting mime
# type so indexer will understand it.
#
# <from_mime> and <to_mime> are standard mime types
# <to_mime> is either text/plain or text/html
#
# Optional charset parameter used to change charset if needed
#
# Command line may have $1 parameter which stands for temporary file name.
# Some parsers can not operate on stdin, so indexer creates temporary file
# for parser and it's name passed instead of $1. Take a look into documentation
# for other parser types and parsers usage explanation.
# Examples:
#
#      from_mime               to_mime[charset]          [command line [$1]]
#
#Mime application/msword      "text/plain; charset=utf-8" "catdoc -a -dutf-8 $1"
#Mime application/msword      "text/html; charset=utf-8"  "wvHtml --charset=utf-8 $1 -"
#Mime application/x-troff-man  text/plain               "deroff"
#Mime text/x-postscript       text/plain               "ps2ascii"
#Mime application/pdf         text/plain               "pdftotext $1 -"
#Mime application/pdf         text/html                "pdftohtml -noframes -enc UTF-8 -i -stdout $1"
#Mime application/vnd.ms-excel text/plain              "xls2csv $1"
#Mime application/vnd.ms-excel text/html               "xlhtml $1"
#Mime "text/rtf*"             text/html                "rthc --use-stdout $1 2>/dev/null"
#Mime "text/rtf*"             text/xml                 "rtfx -w $1 2>/dev/null"
#Mime "text/rtf*"             text/html                "unrtf --html $1"
#Mime application/vnd.ms-powerpoint "text/html; charset=utf-8" "pptohtml $1"
#Mime application/vnd.ms-powerpoint text/html           "ppthtml $1"



#########################################################################
# Parseurs avec sortie au format texte pour indexation par mnoGoSearch
#########################################################################


# Pour documents MS Word *.doc
# Parseur par defaut et recommander
Mime application/msword      "text/plain;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/catdoc-0.94.4/bin/catdoc -m72 -fascii -s cp1252 -d cp1252 $1"


# Pour documents MS Powerpoint 97-2003*.ppt
# ATTENTION catppt dans catdoc-0.94.2 et catdoc-0.94.4 ne fonctionne pas correctement
# avec les fichiers *.ppt créés par MS Powerpoint plus récent que la version 2003 et OpenOffice/LibreOffice
# utilisez ppthtml imperativement ci-dessous
# Mime application/vnd.ms-powerpoint "text/plain;charset=utf-8"   "/opt/mnogosearch_3.3.14/parseurs/catdoc-0.94.4/bin/catppt -s cp1252 -d cp1252 $1"
```

 # Pour documents MS Excel 5,97-2003 *.xls

 # Parseur par defaut et recommander

Mime application/vnd.ms-excel "text/plain;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/catdoc-0.94.4/bin/xls2csv -ccomma -bformfeed -fdmy -s cp1252 -d cp1252 $1"

 # Pour documents Adobe Acrobat *.pdf

 # Mime application/pdf "text/html;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/xpdf-3.03/bin/pdftotext -layout -enc Latin1 -eol unix -q $1 -"

 # Pour documents bureautiques au format RTF comme ceux de MS Wordpad

 # Mime text/rtf "text/plain;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/unrtf-0.21.3/bin/unrtf --nopict --text -P /opt/mnogosearch_3.3.14/parseurs/unrtf-0.21.3/share/unrtf $1"

 # Pour documents StarOffice/OpenOffice Writer *.sxw

 # Mime application/vnd.sun.xml.writer "text/plain;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/sxwtoplain/sxwtoplain.sh $1"

 # Pour documents OpenOffice/LibreOffice Writer *.odt à la norme ODF (Open Document Format)

 # Mime application/vnd.oasis.opendocument.text "text/plain;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/odttoplain/odttoplain.sh $1"

 # Pour documents MS Word *.docx (a partir de MS Word 2007) a la norme Office Open XML

 # Mime application/application/vnd.openxmlformats-officedocument.wordprocessingml.document "text/plain;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/docxtoplain/docxtoplain.sh $1"

 # Pour documents MS Word *.docx (a partir de MS Word 2007) a la norme Office Open XML

 # Parseur par defaut et recommander

Mime application/application/vnd.openxmlformats-officedocument.wordprocessingml.document "text/plain;charset=ISO-8859-1" "/opt/mnogosearch_3.3.14/parseurs/docx2txt-1.2/bin/docx2txt.pl $1 -"

 # Pour documents MS Powerpoint *.pptx (a partir de MS Powerpoint 2007) a la norme Office Open XML

 # Parseur par defaut et recommander

Mime application/application/vnd.openxmlformats-officedocument.presentationml.presentation "text/plain;charset=ISO-8859-1" "/opt/mnogosearch_3.3.14/parseurs/pptx2txt-0.1/bin/pptx2txt.pl $1 -"

 # Pour documents MS Excel *.xlsx (a partir de MS Excel 2007) a la norme Office Open XML

 # Unique parseur pour fichiers *.xlsx

Mime application/vnd.openxmlformats-officedocument.spreadsheetml.sheet "text/plain;charset=ISO-8859-1" "/opt/mnogosearch_3.3.14/parseurs/xlsx2csv-0.11/bin/xlsx2csv.py $1"

 # Pour documents GhostScript *.ps

 # Unique parseur pour fichiers *.ps

Mime application/postscript "text/plain;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/ghostscript-9.07/bin/ps2ascii $1"

 # Pour documents WordPerfect *.wpd

 # Mime application/vnd.wordperfect "text/plain" "/opt/mnogosearch_3.3.14/parseurs/libwpd-0.9.6/bin/wpd2text $1"

 # Pour documents MS Works *.wps

 # Mime application/vnd.ms-works "text/plain" "/opt/mnogosearch_3.3.14/parseurs/libwps-0.2.7/bin/wps2text $1"

###

 # Parseurs avec sortie au format HTML pour indexation par mnoGoSearch

###

 # Pour documents MS Powerpoint 97-2003 *.ppt

 # Parseur par defaut et recommander

Mime application/vnd.ms-powerpoint "text/html" "/opt/mnogosearch_3.3.14/parseurs/xlhtml-0.5.1/bin/ppthtml $1"

 # Pour documents MS Excel 5,97-2003 *.xls

 # Mime application/vnd.ms-excel "text/html" "/opt/mnogosearch_3.3.14/parseurs/xlhtml-0.5.1/bin/xlhtml $1"

 # Pour documents bureautiques au format RTF comme ceux de MS Wordpad

 # Parseur par defaut et recommander

 # Mime text/rtf "text/html;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/unrtf-0.21.3/bin/unrtf --nopict --html -P /opt/mnogosearch_3.3.14/parseurs/unrtf-0.21.3/share/unrtf $1"

 # Pour documents Adobe Acrobat *.pdf

 # Xpdf avec sortie HTML / recommander si necessiter sortie en HTML

 # Parseur par defaut et recommander

Mime application/pdf "text/html;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/xpdf-3.03/bin/pdftotext -layout -enc Latin1 -eol unix -q -htmlmeta $1 -"

 # Poppler avec sortie HTML

Mime application/pdf "text/html;charset=ISO-8859-1" "/opt/mnogosearch_3.3.14/parseurs/poppler-0.22.1/bin/pdftohtml -i -noframes -stdout -nomerge -enc Latin1 -nodrm $1"

 # Pour documents StarOffice/OpenOffice Writer *.sxw

 # Parseur par defaut et recommander

Mime application/vnd.sun.xml.writer "text/html;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/soffice2html-0.78/bin/soffice2html.pl -w -O $1"

 # Pour documents OpenOffice/LibreOffice Writer *.odt

 # Parseur par defaut et recommander

 # Mime application/vnd.oasis.opendocument.text "text/html;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/odftools-0.1/bin/odf2html $1"

 # Pour documents WordPerfect *.wpd

 # Parseur par defaut et recommander

Mime application/vnd.wordperfect "text/html;charset=utf-8" "/opt/mnogosearch_3.3.14/parseurs/libwpd-0.9.6/bin/wpd2html $1"

 # Pour documents MS Works *.wps

 # Parseur par defaut et recommander

Mime application/vnd.ms-works "text/html" "/opt/mnogosearch_3.3.14/parseurs/libwps-0.2.7/bin/wps2html $1"

#

 # decompression de documents compresses dans un certain type d archive

#

#Mime application/zip "text/plain" ""/opt/mnogosearch_3.3.14/parseurs/mnogosearch-gw/mnogosearch-gw -r -l /var/mnogosearch_3.3.14/parseurs/log/mnogosearch-gw-logs.err -s $1"

Use ParserTimeOut to specify amount of time for parser execution

to avoid possible indexer hang.

ParserTimeOut 300

. . .

```
#####################################################################
# Document sections.
#
# Format is:
#
#   Section <string> <number> <maxlen> [clone] [sep] [{expr} {repl}]
#
# where <string> is a section name and <number> is section ID
# between 0 and 255. Use 0 if you don't want to index some of
# these sections. It is better to use different sections IDs
# for different documents parts. In this case during search
# time you'll be able to give different weight to each part
# or even disallow some sections at a search time.
# <maxlen> argument contains a maximum length of section
# which will be stored in database.
# "clone" is an optional parameter describing whether this
# section should affect clone detection. It can
# be "DetectClone" or "cdon", or "NoDetectClone" or "cdoff".
# By default, url.* section values are not taken in account
# for clone detection, while any other sections take part
# in clone detection.
# "sep" is an optional argument to specify a separator between
# parts of the same section. It is a space character by default.
# "expr" and "repl" can be used to extract user defined sections,
# for example pieces of text between the given tags. "expr" is
# a regular expression, "repl" is a replacement with $1, $2, etc
# meta-characters designating matches "expr" matches.

# Standard HTML sections: body, title

Section         body                            1               256
Section title                           2               128

# META tags
# For example <META NAME="KEYWORDS" CONTENT="xxxx">
#

Section meta.keywords                   3               128
Section         meta.description        4               128

# HTTP headers example, let's store "Server" HTTP header
#
#
#Section header.server                  5               64


# Document's URL parts
```

```
Section url.file        6       0
Section url.path                7       0
Section         url.host        8       0
Section url.proto               9       0

# CrossWords

Section crosswords              10      0

#
# If you use CachedCopy for smart excerpts (see below),
# please keep Charset section active.
#
Section Charset                 11      32

Section Content-Type            12      64
Section Content-Language 13     16

# Uncomment the following lines if you want tag attributes
# to be indexed

#Section attribute.alt          14      128
#Section attribute.label 15     128
#Section attribute.summary 16   128
#Section attribute.title 17     128
#Section attribute.face         27      0

# Message/rfc822 headers

Section msg.from                18      0
Section msg.to                  19      0
Section msg.subject             20      0

# Uncomment the following lines if you want use NewsExtensions
# You may add any Newsgroups header to be indexed and stored in urlinfo table

#Section References             18      0
#Section Message-ID             19      0
#Section Parent-ID              20      0

# Uncomment the following lines if you want index MP3 tags.
#Section MP3.Song       21      128
#Section MP3.Album      22      128
#Section MP3.Artist     23      128
#Section MP3.Year       24      128

# Comment this line out if you don't want to store "cached copies"
# to generate smart excerpts at search time.
# Don't forget to keep "Charset" section active if you use cached copies.
# NOTE: 3.2.18 has limits for CachedCopy size, 32000 for Ibase and
```

```
# 15000 for Mimer. Other databases do not have limits.
# If indexer fails with 'string too long' error message then reduce
# this number. This will be fixed in the future versions.
#
Section CachedCopy                      25 64000

# A user defined section example.
# Extract text between <h1> and </h1> tags:
#Section h1                             26 128 "<h1>(.*)</h1>" $1

...

#############################################################################
#Server [Method] [SubSection] <URL> [alias]
# This is the main command of the indexer.conf file. It's used
# to describe web-space you want to index. It also inserts
# given URL into database to use it as a start point.
# You may use "Server" command as many times as a number of different
# servers or their parts you want to index.
#
# "Method" is an optional parameter which can take on of the following values:
# Allow, Disallow, CheckOnly, HrefOnly, CheckMP3, CheckMP3Only, Skip.
#
# "SubSection" is an optional parameter to specify server's subsection,
# i.e. a part of Server command argument.
# It can take the following values:
# "page" describes web space which consists of one page with address <URL>.
# "path" describes all documents which are under the same path with <URL>.
# "site" describes all documents from the same host with <URL>.
# "world" means "any document".
# Default value is "path".
#
# To index whole server "localhost":
#Server http://localhost/
#
# You can also specify some path to index subdirectory only:
#Server http://localhost/subdir/
#
# To specify the only one page:
#Server page http://localhost/path/main.html
#
# To index whole server but giving non-root page as a start point:
#Server site http://localhost/path/main.html
#
#
# You can also specify optional parameter "alias". This example will
# index server "http://www.mnogosearch.org/" directly from disk instead of
# fetching from HTTP server:
```

#Server http://www.mnogosearch.org/ file:///home/httpd/www.mnogosearch.org/

#

 # Indiquer URL du site a indexer ici

 # http://www.monsite.com

 # Indiquer URL a exclure ici

#Disallow *.toto

. . .

###

#URL http://localhost/path/to/page.html

This command inserts given URL into database. This is useful to add

several entry points to one server. Has no effect if an URL is already

in the database. When inserting indexer does not executes any checks

and this URL may be deleted at first indexing attempt if URL has no

correspondent Server command or is disallowed by rules given in

Allow/Disallow commands.

#

#This command will add /main/index.html page:

#URL http://localhost/main/index.html

 # Indiquer URL particuliere du site a indexer ici:

 # http://www.monsite.com/plan.htm

Annexe 2
Grille QSOS d'évaluation de mnoGoSearch

mnoGoSearch 3.3

Information

- **Language:** fr
- **Application:** mnoGoSearch
- **Release:** 3.3
- **License:** GNU GPL
- **URL:** http://www.mnogosearch.org
- **Description:** Moteur de recherche multilingue pour sites web Internet et Intranet
- **Author(s) of this sheet:** Yannick LE NY (yleny@nospam@laposte.net)

  You can access changelog in the CVS.

Maturité

- **Patrimoine**
  - **Age du projet**
    - Inférieur à trois mois
    - Entre trois mois et trois ans
    - **Supérieur à trois ans**
      La première version publique de mnoGoSearch la 1.5 et date du 18/11/1998 soit 14 ans

    Score: 2/2
  - **Historique**
    - Le logiciel connaît de nombreux problèmes qui peuvent être rédhibitoires
    - Pas de problèmes majeurs, ni de crise ou historique inconnu
    - **Bon historique de gestion de projet et de crise**
      Développement un peu ralenti actuellement mais régulier.

    Score: 2/2
  - **Equipe de développement**
    - Très peu de développeurs identifiés ou développeur unique
    - **Quelques développeurs actifs**
    - Equipe de développement importante et identifiée
      5 développeurs coeur plus 9 développeurs qui ne sont plus actifs et 47 contributeurs identifiés. L'équipe coeur est stabilisée par le fait qu'elle soit employée par Lavtech.Com Corp.

    Score: 1/2

- **Popularité**
    - Très peu d'utilisateurs identifiés
    - Usage décelable
    - **Nombreux utilisateurs et références**
      Mnogosearch est un des moteurs de recherche open source les plus populaires

  Score: 2/2
-

- **Activité**
  -

    Communauté des contributeurs
    - Pas de communauté ou de réelle activité (forum, liste de diffusion…)
    - Communauté existante avec une activité notable
    - **Communauté forte : grosse activité sur les forums, de nombreux contributeurs et défenseurs**
      Environ 47 contributeurs différents annoncés par le projet en dehors de l'équipe des développeurs.

    Score: 2/2
  - **Activité autour des bugs**
    - Réactivité faible sur le forum ou sur la liste de diffusion, ou rien au sujet des corrections de bugs dans les notes de versions
    - **Activité détectable mais sans processus clairement exposé, temps de résolution long**
    - Forte réactivité, basée sur des rôles et des assignations de tâches
      Réactivité moyenne avec identification de l'état des corrections

    Score: 1/2
    - **Pas de processus de gestion de bugs identifiés**
    -
    -
  - **Activité autour des fonctionnalités**
    - Pas ou peu de nouvelles fonctionnalités
    - **Évolution du produit conduite par une équipe dédiée ou par des utilisateurs, mais sans processus clairement exposé**
    - Les requêtes pour les nouvelles fonctionnalités sont clairement outillées, feuille de route disponible
      L'outil de gestion de bugs permet également de saisir des demandes de nouvelles fonctionnalités.

    Score: 1/2
  - **Activité sur les releases/versions**
    - Très faible activité que ce soit sur les versions de production ou de développement (alpha, beta)

- **Activité que ce soit sur les versions de production ou de développement (alpha, beta), avec des versions correctives mineures fréquentes**
- Activité importante avec des versions correctives fréquentes et des versions majeures planifiées liées aux prévisions de la feuille de route
  Nouvelles versions publiées régulièrement tous les 2 à 3 mois.

Score: 1/2

- **Gouvernance**
  - **Détenteur des droits**
    - **Les droits sont détenus par quelques individus ou entités commerciales**
    - Les droits sont détenus par de nombreux individus de façon homogène
    - Les droits sont détenus par une entité légale, une fondation dans laquelle la communauté a confiance (ex: FSF, Apache, ObjectWeb)
      L'intégralité des droits est détenue LavTech.Com Corp. Et les contributeurs.

    Score: 0/2
  - **Feuille de route**
    - Pas de feuille de route publiée
    - **Feuille de route sans planning**
    - Feuille de route versionnée, avec planning et mesures de retard
      Les fonctionnalités à implémenter ou bogues à corriger sont détaillés dans l'outil de gestion de bugs. De plus, il existe une TODO list qui indique les fonctionnalités à implémenter sans dates prévisionnelles.

    Score: 1/2
  - **Pilotage du projet**
    - Pas de pilotage clair du projet
    - **Pilotage dicté par un seul individu ou une entité commerciale**
    - Indépendance forte de l'équipe de développement, droits détenus par une entité reconnue
      Les membres et les fonctions de l'équipe dirigeante sont connus et clairement définis,Il s'agit des 5 développeurs principaux de mnogosearch.org. LavTech.Com Corp, unique détenteur des droits et employeur des développeurs principaux et en particulier Alexander Barkov, décide de l'orientation des développements.

    Score: 1/2
  - **Mode de distribution**

- Existence d'une distribution commerciale ou propriétaire ou distribution libre limitée fonctionnellement
- Sous-partie du logiciel disponible sous licence propriétaire (Coeur / Greffons...)
- **Distribution totalement ouverte et libre**
  Le logiciel est totalement libre pour UNIX/Linux/BSD, par contre les binaires Windows sont sous licence commerciale.

Score: 2/2

- **Industrialisation**
  - **Services**
    - Pas d'offre de service identifiée
    - **Offre existante mais restreinte géographiquement ou en une seule langue ou fournie par un seul fournisseur ou sans garantie**
    - Offre riche, plusieurs fournisseurs, avec des garanties de résultats
      Pas de programme de formation, par contre il existe une documentation et un forum pour les questions. Support payant proposé par LavTech.Com Corp et support sur les forums Conseil et aide payants proposé par LavTech.Com Corp, par contre il existe une documentation et un forum pour les questions.

    Score: 1/2
  - **Documentation**
    - Pas de documentation utilisateur
    - La documentation existe mais est en partie obsolète ou restreinte à une seule langue ou peu détaillée
    - **Documentation à jour, traduite et éventuellement adaptée à différentes cibles de lecteurs (end-user, sysadmin, manager…)**
      Documentation complète et à jour. Une version française de la documentation est cours de préparation. Le site web officiel du logiciel est en anglais et il existe un site non officiel complet en français.

    Score: 2/2
  - **Méthode qualité**
    - Pas de processus qualité identifié
    - Processus qualité existant, mais non formalisé ou non outillé
    - **Processus qualité basé sur l'utilisation d'outils et de méthodologies standards**
      Existence d'une test suite intégrée (make check). les bogues de régression sont à soumettre via le site web dans l'outil de gestion des bogues. Outil de suivi des bogues et des demandes de fonctionnalités existant et complet (statistiques, recherche). cf. http://www.mnogosearch.org/bugs/

    Score: 2/2

- **Modification du code**
  - Pas de moyen pratique de proposer des modifications de code
  - **Des outils sont fournis pour accéder et modifier le code (ex : CVS, SVN) mais ne sont pas vraiment utilisés pour développer le produit**
  - Le processus de modification de code est bien défini, exposé et respecté, basé sur des rôles bien définis
    Code source disponible mais pas de guide de développement ou de document d'architecture applicative.

  Score: 1/2
-

Fonctionnalités d'indexation de récupération du robot d'indexation (crawler)

- Description du groupe de critères
  - **Support standard d'exclusion des robots**
    - Non
    - Partiel
    - **Oui**
      "Fichier robot.txt file et tags HTML <meta name= ""robots"" content= ""nofollow""> et <meta name= ""robots"" content= ""noindex"">"

    Score: 2/2
  - **Protocoles supportés**
    - Non
    - Partiel
    - **Oui**
      "HTTP 1.0 et 1.1, FTP, NNTP (News) en natif. HTTPS suivant compilation. Nécessite de compiler mnoGoSearch avec OpenSSL pour le HTTPS"

    Score: 2/2
  - **Support des témoins de connexion (cookies)**
    - Non
    - Partiel
    - **Oui**
      Oui Parametre UseCookie dans fichier indexer.conf

    Score: 2/2
  - **Détection des doublons (documents ou contenu)**
    - Non
    - Partiel
    - **Oui**
      Oui Parametre DetectClones dans fichier indexer.conf

    Score: 2/2
  - **Connecteurs supplémentaires à des sources de données**
    - Non

- Partiel
- **Oui**
  Oui pour bases de données Indexation des champs des tables dans les bases de données suivantes: MySQL, PostgreSQL, SQLite, iODBC, unixODBC, EasySoft ODBC-ODBC bridge, Mimer, Virtuoso, Interbase, Oracle , MS SQL, DB2 , Sybase, InterSystems Cache. Nécessite de compiler mnoGoSearch avec le support d'un ou plusieurs de ces SGBD

Score: 2/2

- **Contrôle de la profondeur de recherche du robot d'indexation (crawler)**
  - Non
  - Partiel
  - **Oui**
    Oui

  Score: 2/2
- **Indexation de serveur sécurisé**
  - Non
  - Partiel
  - **Oui**
    Oui, HTTPS, auhentification basique, LDAP, proxy Authenfication via : HTTP 1.0 basique avec login/mot de passe (via apache htpassword), LDAP avec Apache mod_ldap. Support: pour passer par un proxy, connexion à site en HTTPS si Mnogosearch compilé avec OpenSSL pour ce dernier .

  Score: 2/2
- **Segmenteur de mots dans une phrase (pour langue asiatique CJK)**
  - Non
  - Partiel
  - **Oui**
    Oui Support du Japonais avec Mecab à compiler avec Mnogosearch, et du chinois chinois via une liste dans un fichier fourni avec les sources de mnoGoSearch. Pas de support pour le coréen ancien qui est sans espace ou ponctuation. Le segmenteur est nécessaire pour identifier qu'une suite de caractères forme un mot puisqu'il n'y a pas d'espaces pour l'indiquer. Paramètre LoadChineseList ou LoadThaiList dans fichier indexer.conf

  Score: 2/2
- **Support d'une liste de mots vides (Stopword list)**
  - Non
  - Partiel
  - **Oui**
    Oui Pour plusieurs langues: Catalan, Tchèque, Danois, Allemand, Anglais, Espagnol, Français, Hongrois, Italien, Japonais, Lituanien, Néerlandais, Norvégien, Polonais, Portugais, Russe, Suédois, Slovaque, Turc, Ukrainien, Chinois. Paramètre Include stopwords.conf dans fichier indexer.conf

  Score: 2/2
- **Formats de fichiers indexés (web et bureautiques)**

- Non
- Partiel
- **Oui**
  Pour HTML, XML , PHP, JSP, ASP, MP3 (avec serveur en HTTP1.1pour ce dernier) et texte en natif Support des formats bureautiques suivants via parsers externes : PDF, RTF DOC,XLS,PPT , PS, SXW, ODF.Support des formats de compression suivants via script Perl et décompresseur externe :7z, ZIP, GZIP, BZIP2 ,TAR, RAR, CAB, ISO, ARJ, LZH, CHM, Z, CPIO, RPM, DEB et NSIS. Parsers externes : PDF avec XPDF, RTF avec UnRTF, DOC,XLS,PPT avec Catdoc, ps avec Ghostscript,SXW avec SofficeToHtml, ODF avec ODFReader Fichiers compressés avec p7zip .Note: on peut ajouter autant de parser que l'on veux dans le fichier indexer.conf.Il suffit ensuite de déclarer un type mime et de l'associer à un parseur donné.

Score: 2/2

- **Support des langues internationales (hors anglais)**
  - Non
  - Partiel
  - **Oui**
    Oui, langues européennes et asiatiques avec arabe, japonais, chinois Toutes les langues supportés par Unicode UTF-8 soit jusqu'à 650 actuellement

  Score: 2/2
- **Detection de la langue du document indexé**
  - Non
  - Partiel
  - **Oui**
    Oui 70 langues . Paramètre Include langmap.conf dans fichier indexer.conf

  Score: 2/2
- **Support des entités HTML**
  - Non
  - **Partiel**
  - Oui
    Oui Exemple : la lettre è en entite HTML s'écrit 'à' dans le code HTML.

  Score: 1/2
-

Mécanisme de recherche

- Description du groupe de critères
  - **Méthode d'indexation**
    - Non
    - Partiel
    - **Oui**
      Inverted Index

    Score: 2/2

- **Pertinence de la notation (Relevance Ranking)**
    - Non
    - Partiel
    - **Oui**
      Word weight

  Score: 2/2
-

Fonctionnalités de recherche

- Description du groupe de critères
    - **Recherche booléenne**
        - Non
        - Partiel
        - **Oui**
          Oui Via les opérateurs et (&), ou (|) et négation (~)

      Score: 2/2
    - **Recherche phrase précise (Phrase Matching)**
        - Non
        - Partiel
        - **Oui**
          Oui Via phrase entre « dans la recherche

      Score: 2/2
    - **Recherche sur les attributs**
        - Non
        - Partiel
        - **Oui**
          Oui Sur document entier, corps du document, titre, mot-clé, description (c'est a dire sur contenu des balises HTML : TITLE, BODY, DESCRIPTION , KEYWORD) ou URL

      Score: 2/2
    - **Recherche par ressemblance/approximative (Fuzzy Search)**
        - Non
        - Partiel
        - **Oui**
          Accents , sous-chaines(sub-strings), avec début ou fin de mot partiel et synonymes, dehyphenate (recherche pour bouche-trou, retourne résultat avec bouchetrou aussi). Pas de support soundex et metaphone.

      Score: 2/2
    - **Recherche insensible aux accents**
        - Non
        - Partiel
        - **Oui**
          Oui Permet d'avoir des résultats à une recherche avec le mot-clé electricite sans accents

      Score: 2/2
    - **Recherche avec racine des mots/lemmatisation (Word Forms)**

- Non
- Partiel
- **Oui**

  Oui Nécessite des fichiers Ispell pour chaque langue.Permet en recherchant le mot gaz de rechercher aussi gazier , gazière ...

Score: 2/2

- **Recherche avec caractère joker (Wild Card)**
  - **Non**
  - Partiel
  - Oui

    Non pour la recherche, mais % utilisable sur la recherche restrictive sur URLs. Pas de support pour la recherche des mots commençant par gaz avec la recherche gaz* ; ou avec la recherche ga? on cherche des mots comme gaz ou gas ou gag. La recherche avec * retourne tous les documents indexés.

  Score: 0/2
- **Recherche de données numériques**
  - **Non**
  - Partiel
  - Oui

    Non Pas de support pour l'indexation de mots comme quantité = 15 d'une certaine manière pour permettre ensuite des recherches du type quantité = 15, quantité > 15 ou quantité < 15

  Score: 0/2
- **Sensibilité à la casse**
  - **Non**
  - Partiel
  - Oui

    Non Pas de support pour recherche sur Voiture avec V en majuscule et sans les documents avec voiture avec un v en minuscule

  Score: 0/2
- **Requête en langage naturel**
  - **Non**
  - Partiel
  - Oui

    Non Ne supporte pas les requêtes sous la forme de phrase ou question.

  Score: 0/2
- **Expression régulière**
  - **Non**
  - Partiel
  - Oui

    Non Pas de recherche possible avec par exemple, utiliser /19[789][0-9] pour trouver les années entre 1970 et 1999.

  Score: 0/2
- **Pertinence paramétrable**
  - Non
  - Partiel

- **Oui**
  Oui Dans le template de recherche (+ ou – vieux, + ou – au debut du doc, +ou – d'occurence du mot recherché dans le doc, - ou + de mots proches, + ou – taille d'un doc)

Score: 2/2

-

Autres fonctionnalités

- Description du groupe de critères
  - **Nombre maximum de pages/documents**
    - Non
    - Partiel
    - **Oui**
      Plusieurs millions

    Score: 2/2
  - **Fonction de tri des résultats de recherche**
    - Non
    - Partiel
    - **Oui**
      Oui Par date, pertinence

    Score: 2/2
  - **Formatage des résultats personnalisable**
    - Non
    - Partiel
    - **Oui**
      Oui Via un fichier HTML servant à l'affichage des résultats

    Score: 2/2
  - **Frontal de recherche**
    - Non
    - Partiel
    - **Oui**
      Oui Via mode CGI avec un template HTML ou via PHP si PHP compilé avec support Mnogosearch.Il existe aussi la possibilité d'utiliser Perl. Le frontal est internationnalisé , il suffit de traduire un fichier texte anglais dans une autre langue pour avoir le formulaire dans sa langue.Actuellement disponible en anglais, français et polonais.

    Score: 2/2
-

Interfaces de programmation

- Description du groupe de critères
  - **Interfaces de programmation**
    - Non
    - **Partiel**
    - Oui

En C Il existe une extension PHP développé en C pour ajouter le support de Mnogosearch dans PHP afin d'avoir un formulaire de recherche en PHP.

Score: 1/2

-

- ## Outillage

- Description du groupe de critères
  - **Outils d'administration**
    - Non
    - **Partiel**
    - Oui

      Aucun pour Unix , tout se fait en ligne de commande. Outil graphique pour la version commerciale sous Windows

    Score: 1/2

    -
-

Visit the QSOS website

Liste des figures[2]

[2] La table des illustrations fait le récapitulatif des tableaux, graphiques, cartes, photographies, figures, dessins, plans, etc., s'ils ne sont pas trop nombreux dans le texte, et en permet le renvoi. Si ces éléments sont nombreux, il est préférable de les regrouper hors texte, en fin de mémoire, et de les traiter séparément : table des figures, table des tableaux, table des cartes, etc. La table donne la liste de toutes les illustrations selon l'ordre où elles sont mentionnées dans le texte. Elle doit donner la numérotation de l'illustration, son titre et le numéro de la page. Pour enlever cette note de bas de page, supprimer l'appel de note ci-dessus.

Liste des tableaux

Un moteur de recherche multilingue (Choix, paramétrage et industrialisation).

Mémoire d'Ingénieur C.N.A.M., Paris 2013

RESUME

L'objectif de ce mémoire est d'identifier un moteur de recherche Open Source et de réaliser son industrialisation afin de le rendre le plus facile à mettre en oeuvre et à exploiter.

Une évaluation de certains moteurs de recherche Open Source disponible sur Internet, via la méthode QSOS, est tout d'abord présentée. Le mémoire aborde ensuite l'industrialisation du moteur de recherche mnoGoSearch qui a été retenu.

En conclusion, les usages possibles, du moteur de recherche mnoGoSearch dans un contexte EDF sont indiqués.

Mots clés : Indexation, Spider, Application, Open Source, logiciel libre, moteur de recherche, Système d'information, industrialisation informatique.

SUMMARY

The objective of this thesis is identify an Opensource search engine and to realize this industrialization in order to make it easy to implement and operate.

An evaluation of some Open Source search engines available on the Internet via QSOS method is presented at first. Then the thesis discusses the industrialization of the mnoGoSearch search engine that has been selected.

In conclusion, the possible uses of the mnoGoSearch search engine in EDF (Electricité de France) context are indicated..

Key words : Indexation, Spider, Application, Open source, Free software, search engine,Information system, IT industrialization.