Knowledge Management

La Diffusion et la Recherche de Connaissances
Adjonction d'un thésaurus à Lucene

OCSIMA / LAURENT Jean-Marc
ocsima@gmail.com
http://ocsima.neuf.fr
Dernière révision le 08 septembre 2004

Table des matières

Les Outils Utilisés

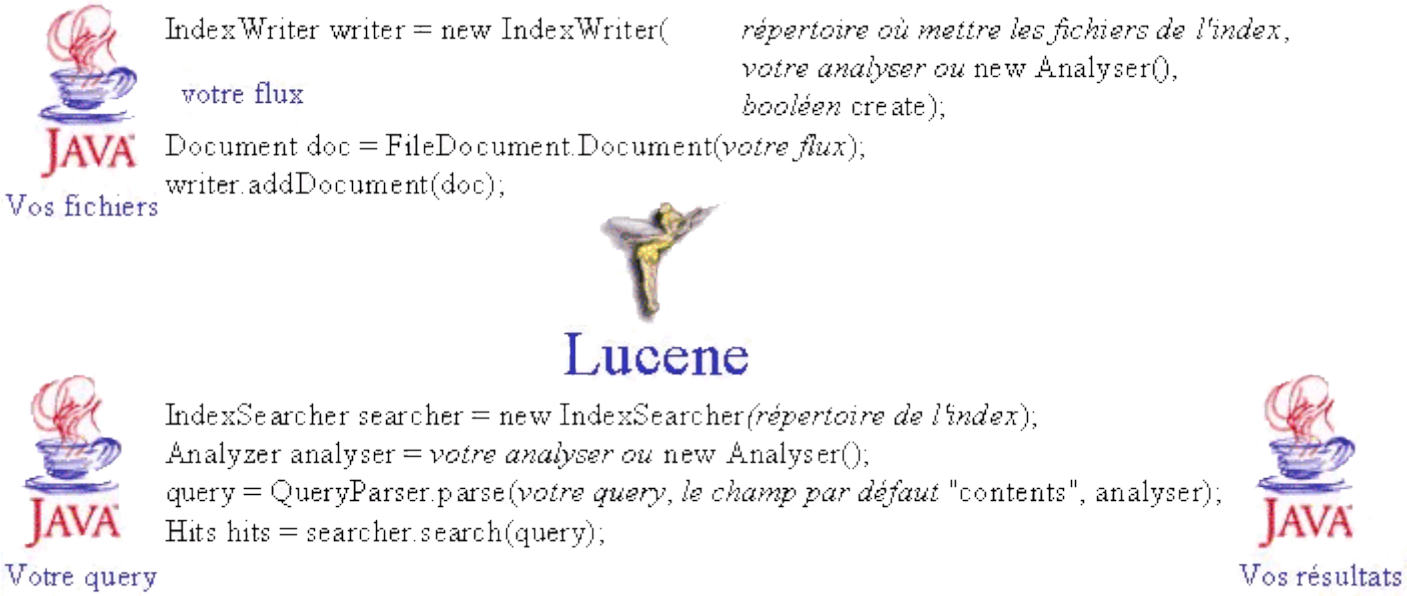
L'application présentée concerne la diffusion, l'indexation et la recherche de documents ou supports de connaissances.

C'est une solution complète et performante :

- les documents seront diffusés, consultés, indexés et recherchés sur un intranet (Tomcat), l'application est portable (langage Java, pages jsp), applets, ...
- l'indexation et la recherche (**Lucene**) se font sur le contenu (plein texte) et sur des meta-données : un formulaire permet leur saisie, vérification et enregistrement au format xml,
- un thésaurus est ajouté, l'exemple est l'analyse physico-chimique : ce thésaurus permet d'enrichir les résultats de la recherche par des données pertinentes non bruitées (synonymie, multi-langage, voisins au sens de noeud dans un arbre xml, ...).

Lucene Qu'est-ce que c'est

Plutôt qu'un long discours un dessin et des captures d'écran :



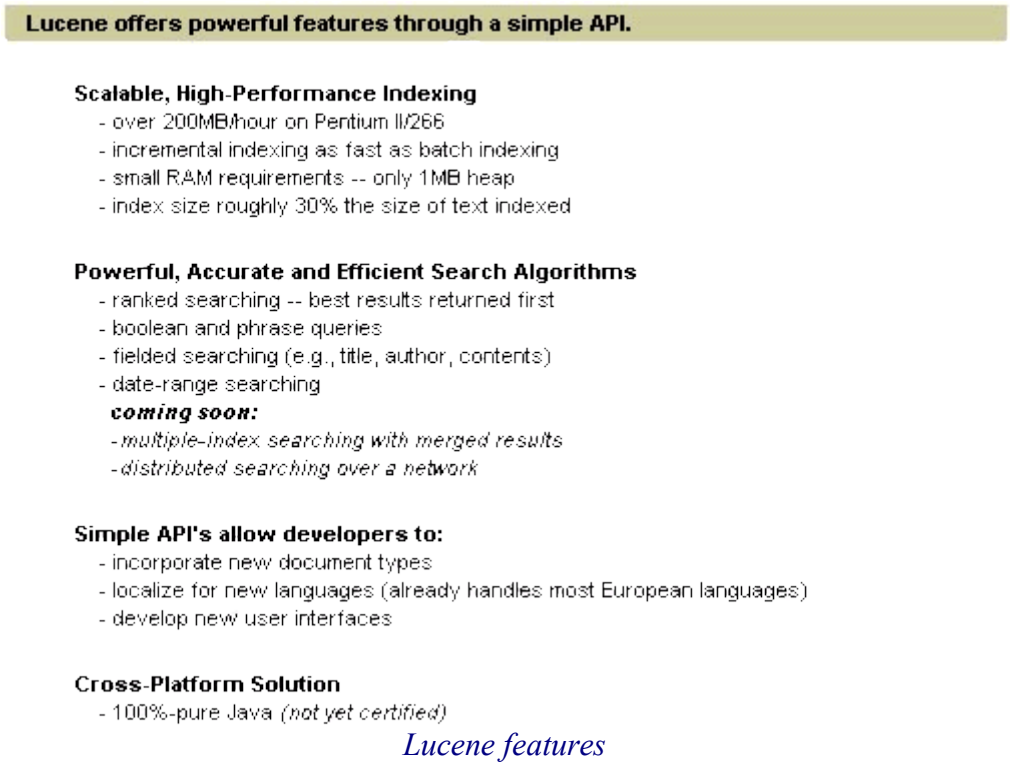
L'application minimale

En regardant un exemple, en une heure vous pouvez développer votre première indexation / recherche. Et ça marche !!!

Doug Cutting

Doug Cutting avait travaillé dans le domaine de la 'recherche d'information' pendant plus de dix ans lorsqu'il a commencé le développement Java de Lucene en 1998, distribué en 'open-source' à partir du printemps 2000.

Ce que l'on peut lire sur le site :



Lucene offers powerful features through a simple API.

Scalable, High-Performance Indexing
- over 200MB/hour on Pentium II/266
- incremental indexing as fast as batch indexing
- small RAM requirements -- only 1MB heap
- index size roughly 30% the size of text indexed

Powerful, Accurate and Efficient Search Algorithms
- ranked searching -- best results returned first
- boolean and phrase queries
- fielded searching (e.g., title, author, contents)
- date-range searching

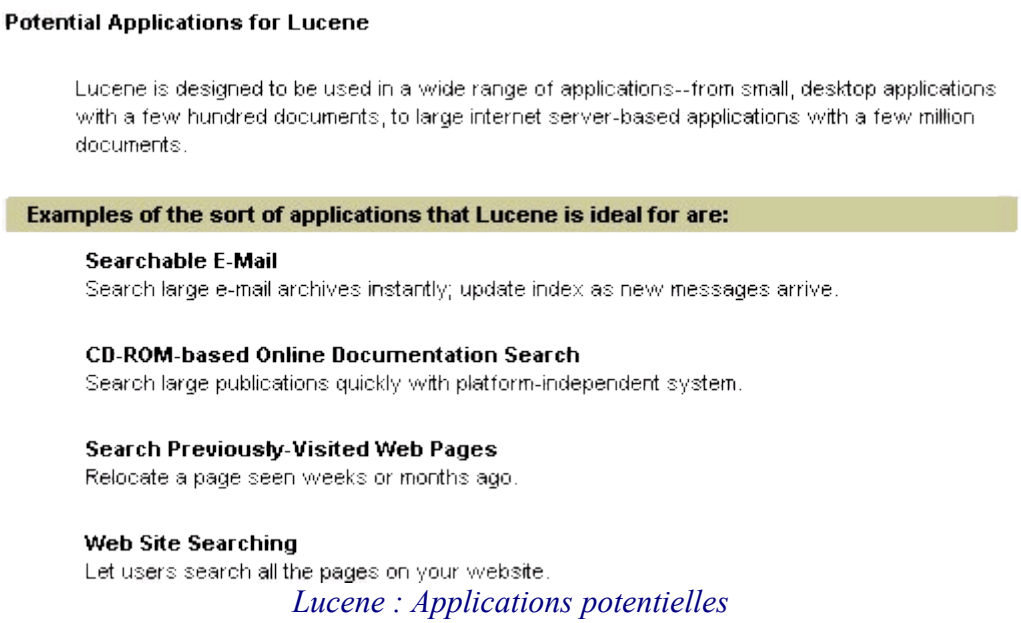
coming soon:
- *multiple-index searching with merged results*
- *distributed searching over a network*

Simple API's allow developers to:
- incorporate new document types
- localize for new languages (already handles most European languages)
- develop new user interfaces

Cross-Platform Solution
- 100%-pure Java *(not yet certified)*

Lucene features

Les applications que l'on peut développer avec Lucene :



Potential Applications for Lucene

Lucene is designed to be used in a wide range of applications--from small, desktop applications with a few hundred documents, to large internet server-based applications with a few million documents.

Examples of the sort of applications that Lucene is ideal for are:

Searchable E-Mail
Search large e-mail archives instantly; update index as new messages arrive.

CD-ROM-based Online Documentation Search
Search large publications quickly with platform-independent system.

Search Previously-Visited Web Pages
Relocate a page seen weeks or months ago.

Web Site Searching
Let users search all the pages on your website.

Lucene : Applications potentielles

La complexité augmente lorsque l'on veut utiliser un dictionnaire et un thésaurus. Il faut mettre les mains dans le cambouis pour intervenir où l'on veut et quand on veut.

Les Documents

HTML est la norme de fait pour la diffusion à contenu statique sur intranet, mais comment l'obtenir : les traitements de texte actuels proposent dans leur menu une sortie HTML.

Le Rêve : OpenOffice.org

Tout le monde utilise la suite bureautique OpenOffice.org. Tout le monde remplit soigneusement les propriétés du fichier. Tout est très facile car tout est enregistré en fait au format XML, ce qui permet l'indexation des différents champs : contenu, titre et aussi auteur, ses coordonnées, ...

La Réalité

Il est impossible d'imposer un outil bureautique commun à un ensemble de rédacteurs. De nombreuses excellentes raisons seront toujours évoquées : "J'ai l'habitude de travailler avec ..., je ne vais pas changer. !" ou "Je sais utiliser telle fonction de ... pour écrire mes formules !" etc. La liste sera longue.

Il vaut mieux prévoir une hétérogénéité maximale.

Quels Formats ?

HTML

Il faut une version au format permettant une visualisation sur l'intranet : le **format HTML**. C'est le format standard et léger. (Le **format PDF** peut toujours être lu, mais il nécessite l'ouverture d'Acrobat Reader.) Les outils bureautiques permettent de réaliser directement une sortie des documents au format html.

TXT

Une version au **format TXT** facilite l'indexation du contenu : il serait possible d'indexer les pages HTML, mais il faudra alors suivre les liens : Microsoft Word pour chaque document .doc sauvegardé au format HTML, crée un répertoire de même nom avec l'extension "_fichier" ou "_file", suivant la version utilisée, qui peut contenir d'autres fichiers HTML, il faut alors suivre les liens. On peut toujours les préciser dans l'algorithme d'indexation, mais pour aujourd'hui pas pour demain ! Les outils bureautiques permettent de réaliser directement une sortie des documents au format txt.

Les Meta Données XML

Pourquoi des Meta-Données ?

Le plus simple est de relater un exemple : je recherchais avec le moteur Google des documents parlant de "dosages potentiométriques" (ne pas oublier de mettre les mots entre guillemets). Parmi les résultats affichés figurait mon site : cette phrase figure effectivement dans mon document de description du Knowledge Management, mais cela ne m'intéressait pas du tout. Je recherchais des documents ayant pour sujet la chimie. Un autre exemple : Je recherche un document écrit par Victor Hugo, mais je ne veux pas de documents parlant de Victor Hugo. Il faut pouvoir spécifier certaines données attachées au document. J'ai développé une interface (Fiche XML) qui permet de les saisir et de les enregistrer au format XML.

Pourquoi le Format XML pour les Meta Données ?

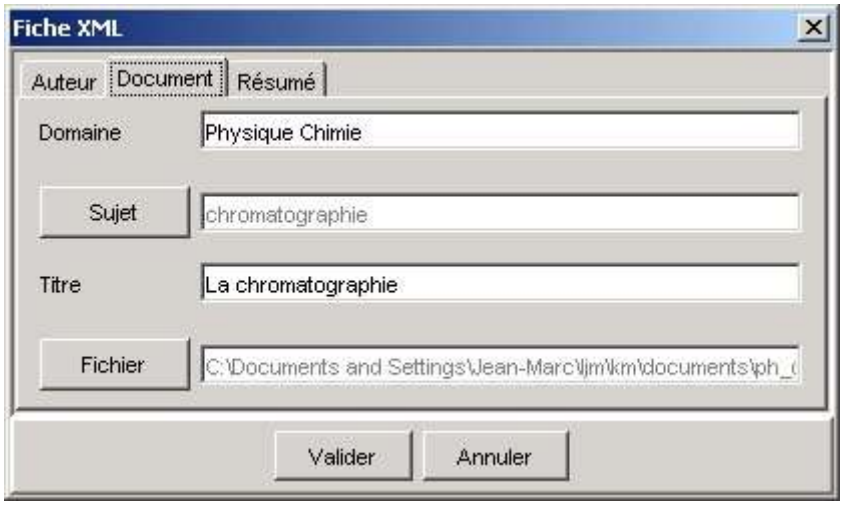
Un document enregistré sous ce format est plus lisible que des champs séparés par des ";" ou des tabulations. Les API Java permettent facilement de manipuler les noeuds de l'arbre obtenu.

Quels Champs Retenir ?

En premier, les coordonnées de l'auteur : elles permettront de le contacter. Elles constituent également la première brique d'un outil de **Recherche d'Expertise** : après avoir lu les documents, l'utilisateur souhaitera peut-être des éclaircissements, des renseignements complémentaires, ... Ces fiches permettent de répondre à la question : Qui est expert pour le domaine considéré ? Qui a publié sur ce sujet ?

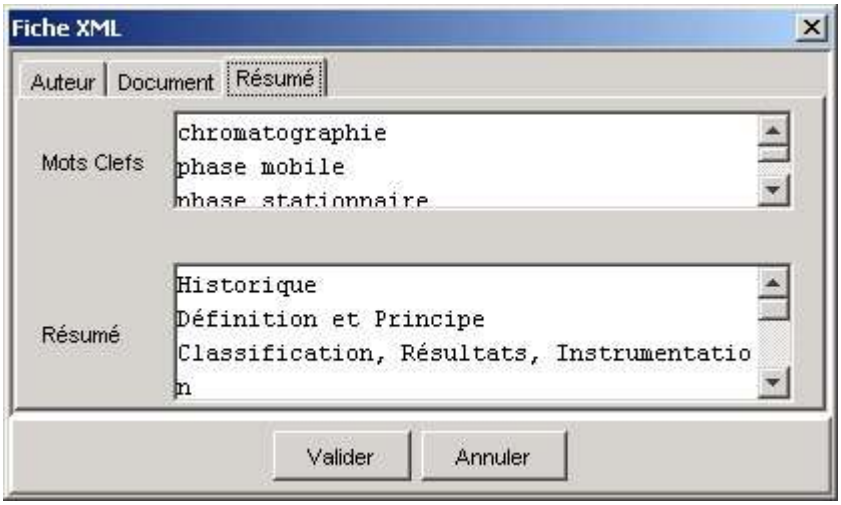
Fiche XML : l'onglet "Auteur"

Ensuite des renseignements concernant le document lui même : le sujet, le domaine (ou sous-sujet), le titre et le nom du fichier (le fichier xml créé portera le même nom). Certaines de ces données seront affichées avec le résultat et permettront la réalisation, par logiciel, d'un index (html) de navigation. La répartition des documents dans des domaines et des sujets est préférable à un "rangement en tas", permet le respect de règles de confidentialité et / ou sécurité, est à adapter pour chaque entreprise. La liste des sujets doit être établie à l'avance et proposée comme liste de choix au rédacteur.



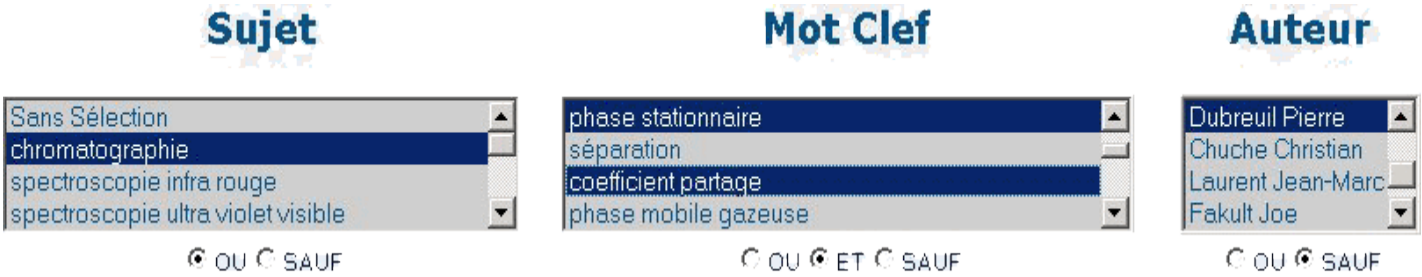
Fiche XML : l'onglet "Document"

Enfin les Mots Clés et le Résumé : ce dernier fera partie du résultat de la recherche, il permettra une sélection plus facile des documents. Une liste de mots-clés sera également proposée à l'utilisateur pour définir sa recherche.



Fiche XML : l'onglet "Résumé"

Lorsque le rédacteur veut valider la fiche, une 'Alert' permet de visualiser les mots qui ne figurent pas dans le thésaurus ou dans le dictionnaire ce qui permet d'apporter les corrections à la fiche ou d'émettre des suggestions concernant le thésaurus.



Les Listes de Choix (pour la Recherche de Documents) sur les Meta Données

La 'query' ci dessus précise que le résultat de la recherche ne doit contenir que des documents dont le sujet est la 'chromatographie', qui ont pour mots-clés 'phase stationnaire' **et** 'coefficient de partage' et dont l'auteur n'est pas 'Dubreuil Pierre'.

Le Dictionnaire et les Mots Vides

N.B. : les lettres accentuées sont maintenues : les mots 'soude' et 'soudé' sont différents.

Un dictionnaire spécifique au domaine doit également être constitué. Il permet de vérifier le contenu des documents indexés et des Fiches XML. Les fautes de frappe peuvent frapper n'importe où !!!

L'Analyser que j'ai défini indexe correctement "informatique" dans "l'informatique" ; un analyser peut à la rigueur corriger des erreurs du type '1Thésaurus', '2Dictionnaire', ... mais ne pourra pas apporter de corrections lorsque la numérotation utilise des lettres : 'aThésaurus', 'bDictionnaire', ...

La définition d'une liste de 'Mots Vides' est également très importante : ces mots vides non significatifs (le, la, les, un, une, des, ...) doivent être éliminés : ils augmentent la taille de l'index (ils apparaissent très souvent) et bruitent les réponses.

Autrefois, l'élimination de ces mots vides posait problème : une recherche sur "pomme de terre" retournait, après l'élimination du mot vide 'de', tous les documents parlant du fruit 'pomme' et de la 'Terre'. Il est maintenant possible, avec Lucene, de spécifier une contrainte de proximité : une recherche sur "pomme terre"~3 ne retournera que les documents où les mots 'pomme' et 'terre' apparaissent à l'intérieur de 3 mots consécutifs.

Actuellement je n'ai établi qu'un seul dictionnaire français, lorsque l'Analyser indexe un texte en langue anglaise, repéré par un nom de fichier terminé par '_en', il ne remplit pas la liste de mots ne figurant pas dans le dictionnaire. Il sera également possible de changer de dictionnaire, et d'en établir un par langue fréquemment utilisée.

L'étape d'indexation et de recherche doivent utiliser le même Analyser.

Le Thésaurus

L'analyse de connaissances nous amène très souvent à considérer l'aspect statique et l'aspect dynamique des entités recensées. Exemple : l'acide acétique est un acide faible. Il peut parfois 'jouer le rôle' de solvant. Cette distinction entre ces deux aspects est très fréquente. Autre exemple pris du langage informatique Java : pour contourner les difficultés inhérentes à 'l'héritage multiple' non disponible, une distinction est faite entre les classes et les interfaces. Une Classe ne peut spécifier qu'une et une seule autre Classe mère dont elle hérite les propriétés (extends) mais peut utiliser et / ou redéfinir les méthodes (implements) d'un nombre quelconque d'Interfaces. Voici ce que l'on peut lire dans le tutorial Java :

> The bicycle class and its class hierarchy defines what a bicycle can and cannot do in terms of its "bicycleness." But bicycles interact with the world on other terms. For example, a bicycle in a store could be managed by an inventory program. An inventory program doesn't care what class of items it manages as long as each item provides certain information, such as price and tracking number. Instead of forcing class relationships on otherwise unrelated items, the inventory program sets up a protocol of communication. This protocol comes in the form of a set of constant and method definitions contained within an interface. The inventory interface would define, but not implement, methods that set and get the retail price, assign a tracking number, and so on.
>
> To work in the inventory program, the bicycle class must agree to this protocol by implementing the interface. When a class implements an interface, the class agrees to implement all the methods defined in the interface. Thus, the bicycle class would provide the implementations for the methods that set and get retail price, assign a tracking number, and so on.
>
> You use an interface to define a protocol of behavior that can be implemented by any class anywhere in the class hierarchy. Interfaces are useful for the following:
>
> - Capturing similarities among unrelated classes without artificially forcing a class relationship.
> - Declaring methods that one or more classes are expected to implement.
> - Revealing an object's programming interface without revealing its class.

Définition et utilisation des Interfaces

Dans le Thésaurus réalisé, c'est cette notion de 'rôle' qui est utilisée.

Les principaux <éléments> utilisés

J'ai retenu, pour le domaine de l'analyse physico-chimique les éléments suivants :

- les substances chimiques (acide, base, alcool, sel, ...),
- les sujets (différents types de potentiométrie, différents types de chromatographie, ...),
- les éléments du matériel (burette, pompe, colonne, système d'injection, ...),
- les termes utilisés (chromophore, longueur d'onde, produit solubilité, ..).

Ces éléments contiennent les 'sous-éléments' suivants, en plus de l'attribut 'fr' = sa désignation en français, :

- en = sa désignation en anglais,
- def = un texte explicatif
- father = élément père (le père de l'acide acétique est 'acide faible' dont le père est 'acide' ...),
- synonyms = la liste des synonymes (ex. : acide éthanoïque).

Les deux derniers 'sous-éléments' sont terms/usingterm et roles/role. Les éléments 'usingterm' et 'role' sont 'EMPTY' mais ont pour attribut 'termid' une référence sur l'id d'un term.

L'élément <role> permet d'implémenter les rôles que les substances chimiques peuvent "jouer".

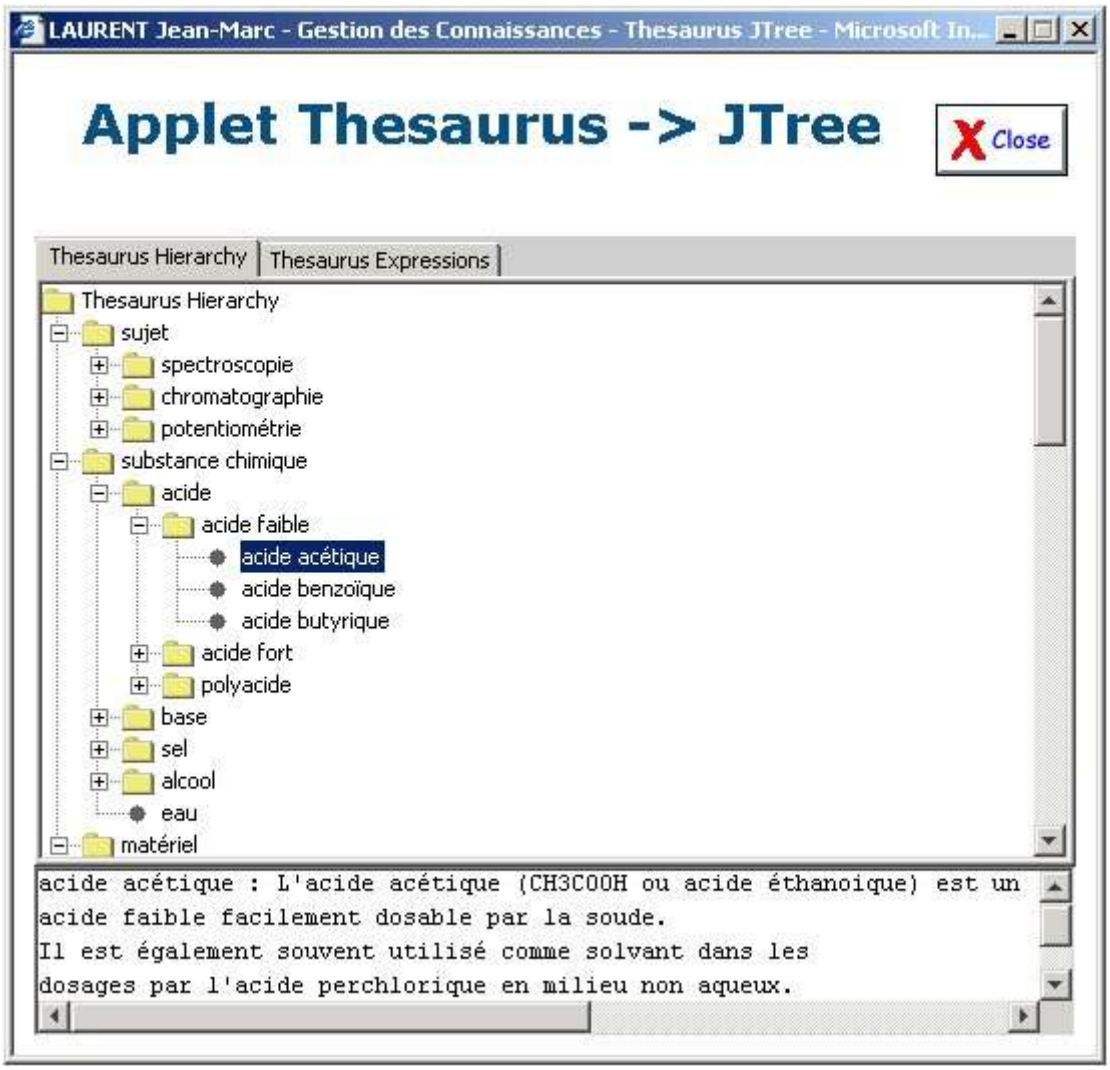
Exemple : l'acide acétique est un 'acide faible' mais il peut avoir le rôle de solvant en 'potentiométrie acide base en milieu non aqueux'. L'acide acétique, l'acide benzoïque et l'acide butyrique sont trois frères dans l'arbre. Mais, comme on le vérifie sur l'applet de visualisation, l'acide acétique peut jouer un rôle que les deux autres ne présentent pas. Lorsqu'on lance une recherche avec la query "acide butyrique" et qu'aucun document indexé ne le mentionne, le système de recherche présente les résultats concernant l''acide benzoïque' (s'il y en a) et non tous les documents contenant 'acide acétique'.

Document Type Definition (DTD)

```
<!ELEMENT thesaurus ( subject | substance | material | term )* >
<!ATTLIST thesaurus domain NMTOKEN #REQUIRED >
<!ATTLIST thesaurus fr CDATA #REQUIRED >
<!ATTLIST thesaurus id ID #REQUIRED >

<!ELEMENT subject ( en | father | def | synonyms | terms )* >
<!ATTLIST subject fr CDATA #REQUIRED >
<!ATTLIST subject id ID #REQUIRED >

<!ELEMENT substance ( en | father | def | formule | roles | synonyms )* >
<!ATTLIST substance fr CDATA #REQUIRED >
<!ATTLIST substance id ID #REQUIRED >

<!ELEMENT material ( en | father | def | synonyms )* >
<!ATTLIST material fr CDATA #REQUIRED >
<!ATTLIST material id ID #REQUIRED >

<!ELEMENT term ( en | father | def | synonyms )* >
<!ATTLIST term fr CDATA #REQUIRED >
<!ATTLIST term id ID #REQUIRED >

<!ELEMENT en ( #PCDATA ) >
<!ELEMENT father EMPTY >
<!ATTLIST father ref IDREF #REQUIRED >
<!ELEMENT def ( #PCDATA ) >
<!ELEMENT formule ( #PCDATA ) >
<!ELEMENT roles ( role+ ) >
<!ELEMENT role EMPTY >
<!ATTLIST role termid IDREF #REQUIRED >
<!ELEMENT synonyms ( synonym+ ) >
<!ELEMENT synonym ( #PCDATA ) >
<!ELEMENT terms ( usingterm+ ) >
<!ELEMENT usingterm EMPTY >
<!ATTLIST usingterm termid IDREF #REQUIRED >
```

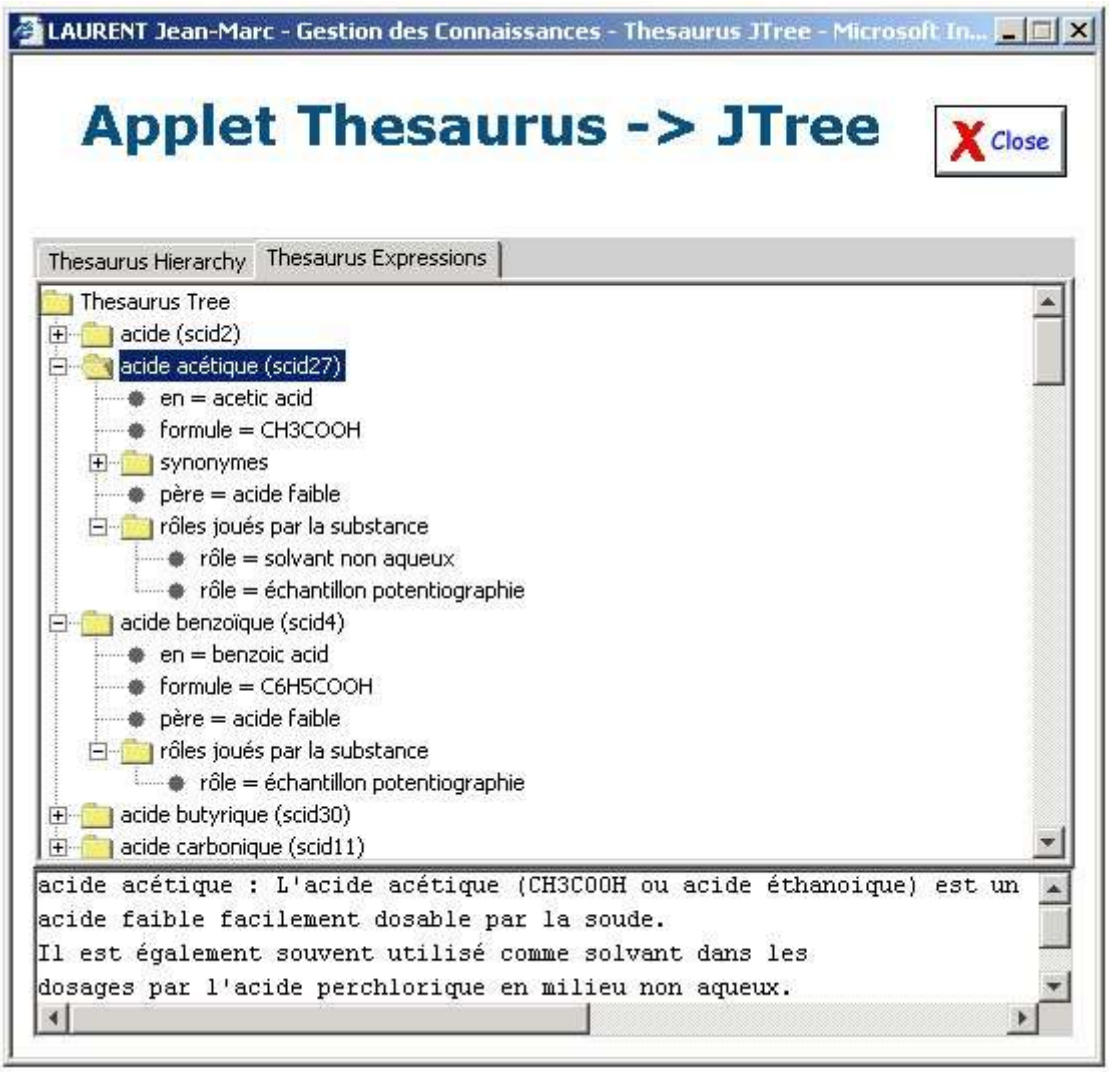
Document Type Definition du Thésaurus

Applet de visualisation du Thésaurus



Le premier onglet de l'Applet

Applet de visualisation du Thésaurus (2ème onglet)



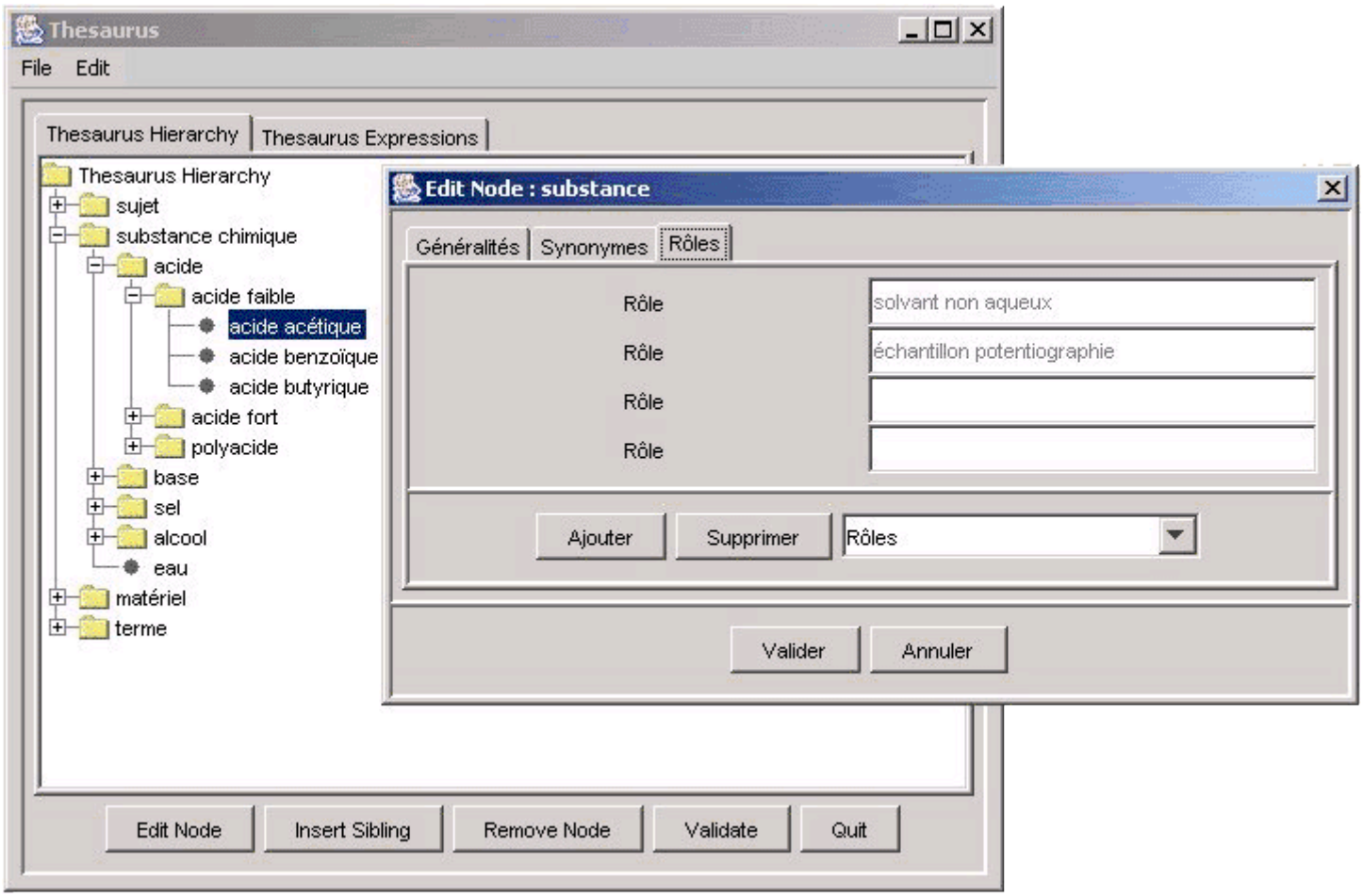
Le second Onglet de l'Applet

La Gestion du Thésaurus

Comme pour les fiches XML, une interface de gestion permet (boutons et items de menus), après avoir sélectionné un noeud de l'arbre, et lorsque cela est autorisé, ;

- d'éditer un noeud (dialogue secondaire comprenant plusieurs onglets, pouvant appeler des dialogues supplémentaires : ajout / retrait de rôles),
- d'ajouter (après initialisation) un noeud frère du noeud sélectionné et de l'éditer,
- d'éliminer le noeud sélectionné,
- de valider le thésaurus ('validating parser'), il est toujours tentant d'utiliser un éditeur quelconque et de modifier, de manière externe, le thésaurus ; il faut donc offrir un moyen de vérifier que ce thésaurus est toujours valide (surtout au niveau des ID et des IDREF, c'est l'intérêt de rassembler tous les éléments du thésaurus dans un même fichier, sinon il faudrait étudier des références externes) par rapport à la DTD,
- et de quitter l'application.

La figure suivante montre la fenêtre principale (toujours l'exemple du noeud <substance fr="acide acétique" ...> ainsi que l'onglet secondaire d'édition des rôles. (Rappel : Un tel noeud ne contient que la référence au terme (une IDREF), il n'est pas possible de l'éditer directement à ce niveau. Pour ajouter un rôle, l'utilisateur doit sélectionner une valeur dans la liste de choix de termes proposée.)
Pour éliminer un 'rôle', il faut commencer par le sélectionner dans la liste.



La fenêtre principale de gestion du thésaurus et l'onglet 'Rôles' du dialogue d'édition

L'Utilisation (utilité) de ces Documents

L'équipe responsable de la conception du Thésaurus et de sa gestion, et l'équipe rédactrice des documents publiés se voient attribuer de lourdes responsabilités et tâches. Dans le cadre d'un projet de Knowledge Management la rétribution de ces dernières doit être établie. Certes le prestige et / ou reconnaissance sous-jacents font partie de cette rétribution, nous allons vérifier toutefois que ces efforts de conception / gestion ne sont pas vains.

Les données des Fiches XML

Affichage des Résultats des Recherches

Le titre et le résumé du document seront affichés dans les résultats des recherches, ce qui permet une meilleure sélection (texte semblant le plus intéressant) parmi les solutions possibles que ne le permet une simple mise en sur brillance des mots de la recherche dans les extraits correspondants. Les coordonnées de l'auteur seront également rappelées.

Réalisation Automatique d'une page Menu de Navigation

En indiquant dans la Fiche XML d'un document le domaine et le sujet auquel il se réfère on rend possible la réalisation par algorithme lors de l'indexation d'une page 'index.html' facilitant la navigation.

Établissement Automatique des Listes de Choix

Lors de l'indexation les listes complètes de tous les auteurs, des domaines et des mots-clefs rencontrés sont construites. Ces listes constituent les listes de choix pour les recherches dans les meta-données.

Recherche d'Expertise

Ces Fiches XML constituent, comme je l'ai déjà indiqué, la première brique d'une Recherche d'Expertise. A partir de ces Fiches, il est possible de définir qui peut être considéré comme expert pour un domaine / sujet donné : le nombre de documents publiés, leur nombre de lectures, ...

Les données du Thésaurus

Aide à la Recherche

Je n'ai introduit que l'attribut 'fr' et l'élément <en>, mais en suivant la même technique, il est possible de réaliser un thésaurus multi-linguiste : 'de', 'sp', .'it', ...

Les éléments <synonyms> et <synonym> apportent une aide dans les deux sens :

- un utilisateur lance la recherche "acide éthanoique" mais aucun document n'utilise ce terme. Il ne figure pas dans l'index. Le thésaurus permet de relancer une recherche sur "acide acétique", 'acide éthanoique' figurant dans les synonymes de la <substance fr=" acide acétique" ;

- un utilisateur lance la recherche "hydroxyde sodium" mais aucun document n'utilise ce terme. Il ne figure pas dans l'index. Le thésaurus permet de relancer une recherche sur "soude", la <substance fr="hydroxyde sodium" ayant dans sa liste de <synonyms> le <synonym>soude</synonym>.

Lorsqu'une recherche n'aboutit sur aucun résultat, la **Thesaurus Help** propose les résultats d'une recherche sur les frères (les noeuds qui ont le même père), et parmi ceux-ci, ne retient que ceux qui peuvent jouer les mêmes 'rôles'(ou un sous-ensemble).

N.B. : Une **Fuzzy Help** est également disponible, elle permet de retrouver dans l'index les mots ou expressions proches (au sens de '**distance de Levenshtein**' entre deux chaînes, ou rapport du nombre minimum de caractères à changer pour passer d'une chaîne à l'autre sur la longueur de la plus courte chaîne). Exemple : cette Fuzzy Help, avec les documents actuellement indexés, permet, lorsqu'on lance la recherche "acide acétique", de retrouver des documents contenant "acid acetic"~2, (c'est à dire les mots 'acetic et 'acid' dans un intervalle de 2 mots, côte à côte, mais sans imposer d'ordre) et un document contenant "acids acetic"~2 (ce document contient l'expression : "formic and acetic acids").

N.B.: Cette **Fuzzy Help** contient également une implémentation faible d'un algorithme du type '**stemming algorithm**' : elle propose les résultats, pour le mot "acétique", de tous les mots "acét*" figurant dans l'index, c'est à dire de tous les mots ayant pour préfixe "acét". Personnellement, dans le cas de la langue française, je n'ai jamais prêté grande confiance à la qualité des résultats obtenus avec ce type d'algorythme : si à une question "mensu-alité" ils peuvent retrouver des documents parlant de "règlements mensu-els" (dans la langue française la voyelle 'u', placée entre une consonne autre que 'q' et une autre voyelle, joue un rôle particulier de pivot), ces algorithmes ne pourront jamais retourner des documents contenant "régler chaque mois".

Case Based Reasoning

Le Case Based Reasoning, forme simplifiée (car intra-domaine) du Raisonnement par Analogie (inter-domaine), est une réponse au souci de '**capitalisation d'expertise**', utilisée principalement dans les Aides au Diagnostic (vous pouvez consulter mon document "**Méthodologie DIABC, Recensement des Cas**") : l'utilisateur décrit la situation courante (la panne constatée) et le système présente à l'utilisateur la solution (résolue antérieurement ou simulée) d'une situation similaire.

La difficulté réside dans cette détermination / mesure de similitude : En quoi deux descriptions sont-elles similaires ?

Une première approche : la comparaison du nombre d'occurrences de mots du thésaurus présents dans les documents étudiés. (Les fameux 'mots vides' ne sont bien sûr pas comptabilisés.)

Suites et Conclusions

A long terme

Je constate et ne peux m'empêcher de regretter l'origine des principaux problèmes dans le domaine de la **Diffusion de Connaissances**: Pourquoi n'y a-t-il pas une utilisation universelle de l'excellente suite bureautique OpenOffice.org ? Vous rédigez un document, vous pouvez demander une sortie en .xml (en fait c'est du XML), en .pdf, en .html, en .doc, en .txt et en .rtf. Les logiciels de Microsoft se retrouveront bientôt dans la position peu enviable de rester les seuls à ne pouvoir ni lire ni exploiter les données des autres systèmes.

L'avenir de XML ?

Nul ne peut prévoir l'avenir du langage XML. Actuellement défendu par SUN et IBM, parfois reconnu par Microsoft, XML est un langage verbeux. L'évolution du prix des mémoires actuelles fait que ce qui pouvait paraître un inconvénient n'en est plus un. En contrepartie, un fichier .xml est 'humainement' lisible ce qui est un avantage certain. "Les paroles s'envolent, mais les écrits restent." De nombreuses traditions sont transmises oralement depuis très longtemps alors qu'aucune machine ne pourrait utiliser aujourd'hui les premiers programmes que j'ai écrits. Par contre, la myriade de programmes lisant, traduisant, transformant, utilisant des fichiers xml vous donne l'impression qu'écrire aujourd'hui en XML, c'est écrire pour l'éternité.

Conclusions à court terme

Le court terme réside dans la prise de conscience du 'papy-boom' et du 'baby-crack' que les entreprises connaissent ou vont connaître : la génération née autour des années 1950 – 1960 va partir à la retraite, emportant avec elle ses connaissances et expériences, ce qui est tout à fait normal et naturel.

Une entreprise doit suivre une politique à plus long terme et doit sauvegarder (et favoriser la transmission de) ces richesses.

J'ai voulu démontrer qu'il est possible de réaliser à faible coût des outils performants pour la Diffusion / Recherche de Connaissances. C'est dans cet esprit que je transmets les résultats de mes expériences et travaux et je suis avide de connaître d'autres expériences similaires.

Suites

Case Based Reasoning (j'ai travaillé sur ce sujet à plusieurs reprises depuis 1987) et Recherche d'Expertise sont les sujets de mes réflexions immédiates. Je réfléchis également au développement d'outils de veille automatisée : agents logiciels (Système Multi Agents) apprenant (Capacité d'Apprentissage) à rechercher, reconnaître et classer (Case Based Reasoning) les nouvelles informations susceptibles d'intéresser l'utilisateur.