

Rapport de projet du cours de Bases de données

Nicolas GAST, Matthieu GUILLAUMIN, Guillaume LAFON

17 mai 2004

Le but du projet est de créer un médiateur pour interroger différentes sources de données, ici des portails web qui référencent des sites sur la base d'un mot-clé. Le thème choisi était celui des jeux. Deux groupes se sont chargés de créer des portails fonctionnant sur l'utilisation d'une taxonomie. Notre travail consistait à créer le médiateur interrogeant ces deux sources, et à implémenter une interface graphique permettant à l'utilisateur d'effectuer des requêtes de manière conviviale, également grâce à l'utilisation de mots-clés et d'une taxonomie. Notre modèle pour cette interface a été celle du site web <http://www.webbrain.com>, où la taxonomie est directement visualisable par l'utilisateur, permettant des requêtes faciles et efficaces. Pour l'implémentation pratique, nous avons eu recours à une application Java, un peu plus austère que celle proposée par webbrain.com, mais fonctionnant sur le même principe.

1 L'interface graphique

L'interface graphique est très simple : la fenêtre principale est subdivisée en deux, la partie inférieure affiche les résultats de la recherche sur le mot-clé courant, constitué d'une liste défilante d'URL. En dessous de cette liste se trouve un espace de saisie de texte où l'utilisateur tape les mot-clé, ainsi qu'un bouton permettant de lancer la recherche.

La partie supérieure est elle-même subdivisée en trois parties, affichant les pères, frères et fils du mot-clé considéré dans la taxonomie. Le mot lui-même apparaît tout en haut de la fenêtre principale. Un simple clic sur un mot apparaissant comme père, frère ou fils du mot-clé courant permet de le sélectionner comme nouveau mot-clé.

2 La base de données

Nous avons utilisé une base de données pour stocker notre taxonomie et implémenter une recherche efficace des mots liés au mot-clé saisi par l'utilisateur. La base de données est constituée d'une table contenant deux champs : père et fils. Cette structure est particulièrement simple et permet une recherche facile des parents du mot-clé. Pour remplir la base, on s'est servi d'un script Perl permettant d'insérer en une seule saisie tous les éléments de la base.

3 L'algorithme de recherche

La recherche des parents du mot-clé se fait via des requêtes sur la base de données. L'algorithme tient compte de la possibilité de boucles dans la taxonomie (qui en pratique ne devraient pas exister) et termine même dans ce cas. Il utilise également la fonction SQL phonex (une description détaillée est disponible à l'adresse http://sqlpro.developpez.com/Soundex/SQL_AZ_soundex.html permettant de corriger une éventuelle faute de frappe de l'utilisateur en se basant sur la prononciation du mot saisi. Cette fonction associe un code à chaque mot en remplaçant des groupes de lettres à l'intérieur du mot par un entier qui code leur prononciation (le détail de l'algorithme utilise dans un premier temps des remplacements de lettres par d'autres produisant un son équivalent, par exemple tous les c suivis d'un e ou s'un i sont remplacés par des s). Elle est adaptée à la prononciation française (il existe naturellement des programmes similaires pour l'anglais; phonex est en fait une adaptation pour la langue française de l'algorithme soundex qui fonctionne avec des mots anglais).

4 L'affichage des URL

Le principe est d'utiliser les bases de données contruites par les deux autres groupes. On utilise en fait plus précisément une vue sur leurs deux bases de données, et une simple requête sur cette vue nous donne le résultat souhaité.

5 L'implémentation en Java

A chaque part du travail décrit dans les sections précédentes correspond grosso modo une class Java. La class MedConnect s'occupe de la connection

à la base de données et des opérations sur celle-ci. Elle est équipée d'une méthode `getFils` qui recherche les fils de la chaîne de caractères prise en argument (via une requête SQL sur la base de données) et les stocke dans un tableau, d'une méthode `getPeres` qui effectue le travail similaire pour les pères, et d'une méthode `getFreres` utilisant les deux méthodes précédentes. Enfin, la méthode `ResultSet` effectue la recherche des URL correspondant au mot-clé choisi sur les bases de données construites par les autres groupes. On lance une connection sur la base en même temps que l'application. Cette connection est refermée quand on quitte l'application. Les class `Mediator` et `Arbre` servent à l'affichage graphique : `Mediator` gère l'affichage de la fenêtre globale et l'interaction avec l'utilisateur (lancer les recherches et rafraîchir l'affichage quand l'utilisateur clique sur le bouton OK), `Arbre` décrit la structure de la partie supérieure de la fenêtre. La classe `ResRecherche` affiche les résultats de recherches d'URL dans la moitié inférieure de la fenêtre. La class `Wa` sert à fermer l'application de façon propre, en se déconnectant de la base au moment de la demande de fermeture de la fenêtre.