École Polytechnique de l'Université de Tours
64, Avenue Jean Portalis
37200 TOURS, FRANCE
Tél. +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Département Informatique

CAHIER DE SPECIFICATION & PLAN DE DEVELOPPEMENT

Projet :	Réalisation d'une base de données de dictionnaires linguistiques	
Emetteur :	DUPRAZ Ludovic	Coordonnées : EPU-DI
Date d'émission :	28/10/2011	

Validation

Nom	Date	Valide (O/N)	Commentaires

Historique des modifications

Version	Date	Description de la modification
00	13/10/2011	Version initiale
01	28/10/2011	Précisions sur les principales fonctions à réaliser

TABLE DES MATIERES

CAHIER DE SPECIFICATION SYSTEME

1. Introduction

Le Laboratoire Ligérien de Linguistique (LLL) de Tours utilise une application sur la prononciation des unités lexicales en anglais britannique contemporain qui regroupe trois dictionnaires existants. Cette application interagit directement avec une base de données qui comprend les informations syntaxiques, lexicales, et morphologiques, ainsi que des données de fréquence, d'usage et de variation. Cette base de données comprend environs 70 000 entrées provenant des trois dictionnaires.

Lors d'un PFE réalisé par Elodie Bacconnet durant l'année 2010-2011, une restructuration complète de la base de données existante a été faite. Il reste maintenant à faire la migration des données dans cette nouvelle base et à développer de nouveaux modules permettant de les exploiter au logiciel existant.

Le présent document est le cahier de spécifications résultant des besoins du LLL.

L'auteur du document est Ludovic Dupraz et son relecteur est Joël Grisward. Durant ce projet, je serai en contact avec Jean-Michel Fournier et Marjolaine Martin du LLL de l'université de Tours.

2. Contexte de la réalisation

2.1. Contexte

Les chercheurs du Laboratoire Ligérien de Linguistique utilisent une application afin de les aider dans leur recherche linguistique. Cette application regroupe les données des dictionnaires suivants :

- The Cambridge English Pronouncing Dictionary
- The Longman Pronouncing Dictionary
- The Macquarie Dictionary

Une nouvelle base de données a été étudiée l'an dernier. Elle est maintenant beaucoup plus flexible et l'on peut y ajouter un dictionnaire beaucoup plus facilement qu'avant si on le souhaite.

Les personnes visées dans ce projet sont donc les chercheurs du LLL qui utilisent souvent ce logiciel car il permet de cumuler les résultats obtenus par trois dictionnaires en même temps.

Cette application devra être améliorée afin de rendre leurs tâches plus faciles.

2.2. Objectifs

Le but de ce projet est de reprendre l'application existante du LLL et d'y insérer la nouvelle base de données. Par la suite, il faudra développer de nouvelles fonctionnalités, notamment des fonctionnalités de traitement de chaînes.

Pour mener à bien ce projet, j'utiliserai les outils mis en places durant le PFE de l'année 2010-2011 à savoir PhpMyAdmin pour la base de données et Xampp pour exécuter l'application.

2.3. Bases méthodologiques

La programmation se fera en Java durant ce projet. La mise en place d'une convention de nommage a été faite (voir document « Convention de nommage »).

La base de données est une base de données MySql.

3. Description générale

3.1. Environnement du projet

Ce projet possède un existant. En effet, il existe un logiciel que les chercheurs du LLL utilisent déjà pour leurs recherches, et c'est ce logiciel que je devrais améliorer. Le logiciel existant a été codé en Java à l'aide de NetBeans et la base de données MySql a été mise en place sur PhpMyAdmin, c'est donc ces outils que j'utiliserai pour l'améliorer. Il faudra donc nécessairement que les utilisateurs disposent d'un JRE (Java Runtime Environment) afin que le programme puisse s'exécuter.

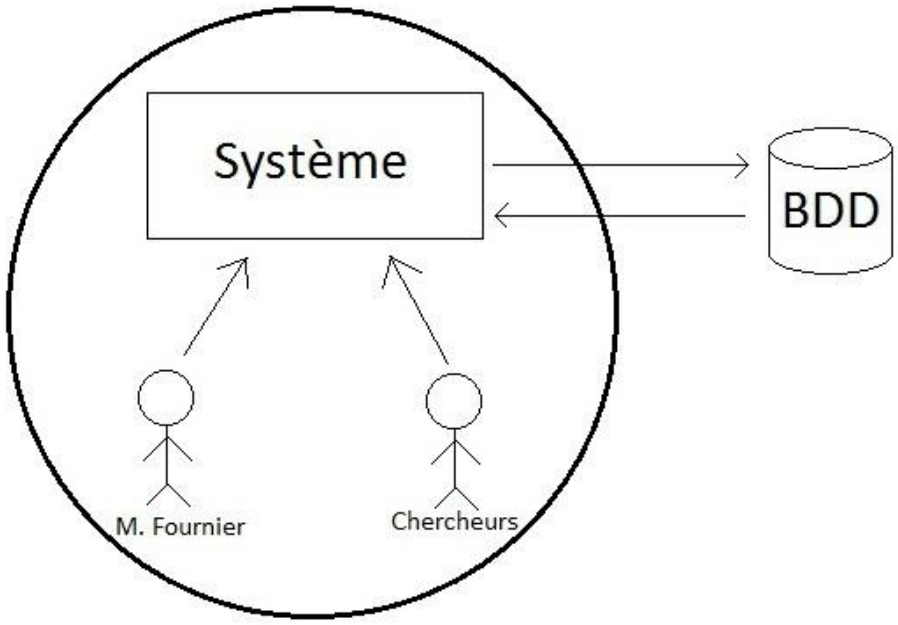


Figure 1 : Diagramme d'environnement

3.2. Caractéristiques des utilisateurs

Les utilisateurs du logiciel seront les chercheurs du LLL et Jean-Michel Fournier. Ils n'utiliseront donc cette application qu'en interne. Ces utilisateurs se servant déjà de l'application, les connaissances requises en informatique ne seront pas plus importantes.

Les chercheurs ne peuvent pas ajouter directement des informations à la base de données. Ils devront soumettre ces dernières à M. Fournier qui lui pourra les sauvegarder dans la base.

Le logiciel est utilisé régulièrement par les chercheurs.

3.3. Contraintes de développement, d'exploitation et de maintenance

3.3.1. Contraintes de développement

Ce projet reprend un projet déjà existant. Il faut donc absolument garder les mêmes outils de développement, à savoir :

- Programmation en Java ;
- Editeur : NetBeans
- Base de données MySql sous PhpMyAdmin
- Kit d'installation Apache : XAMPP
- Fichiers sources de la base de données en XML

3.3.2. Maintenance et évolution du système

Les contraintes liées aux procédures de maintenance sont les même que celles liées au développement.

4. Description des interfaces externes du logiciel

4.1. Interfaces matériel/logiciel

L'interface matériel/logiciel n'est ici pas très conséquente. Un simple ordinateur avec la configuration nécessaire installée (Apache, PhpMyAdminNetBeans,…) et les données des différents dictionnaires utilisés peut exploiter l'application sans problèmes.

Le serveur, depuis lequel on peut modifier la base de données est l'ordinateur de M. Fournier.

4.2. Interfaces homme/machine

L'interface homme/machine a déjà été faite. Il n'y aura que quelques changements à apporter. Voici quelques captures d'écran de l'interface actuelle :

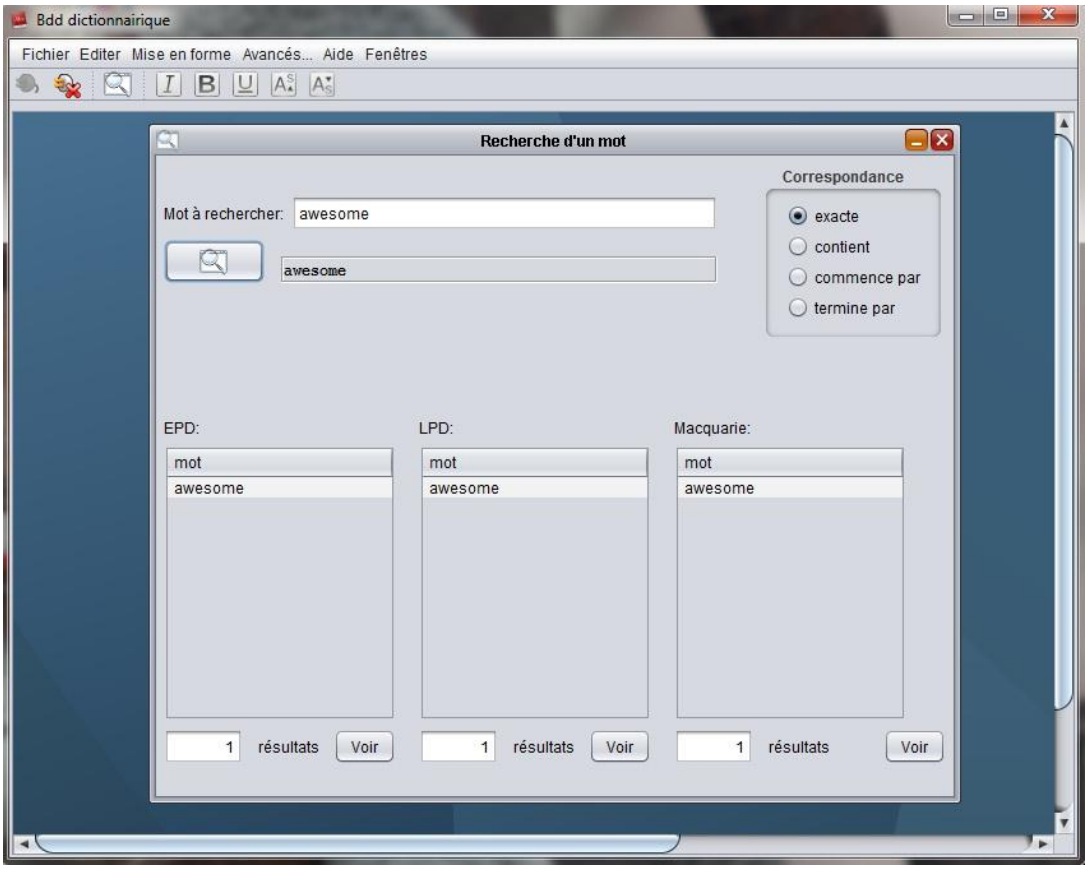


Figure 2 : Interface globale et recherche de mot

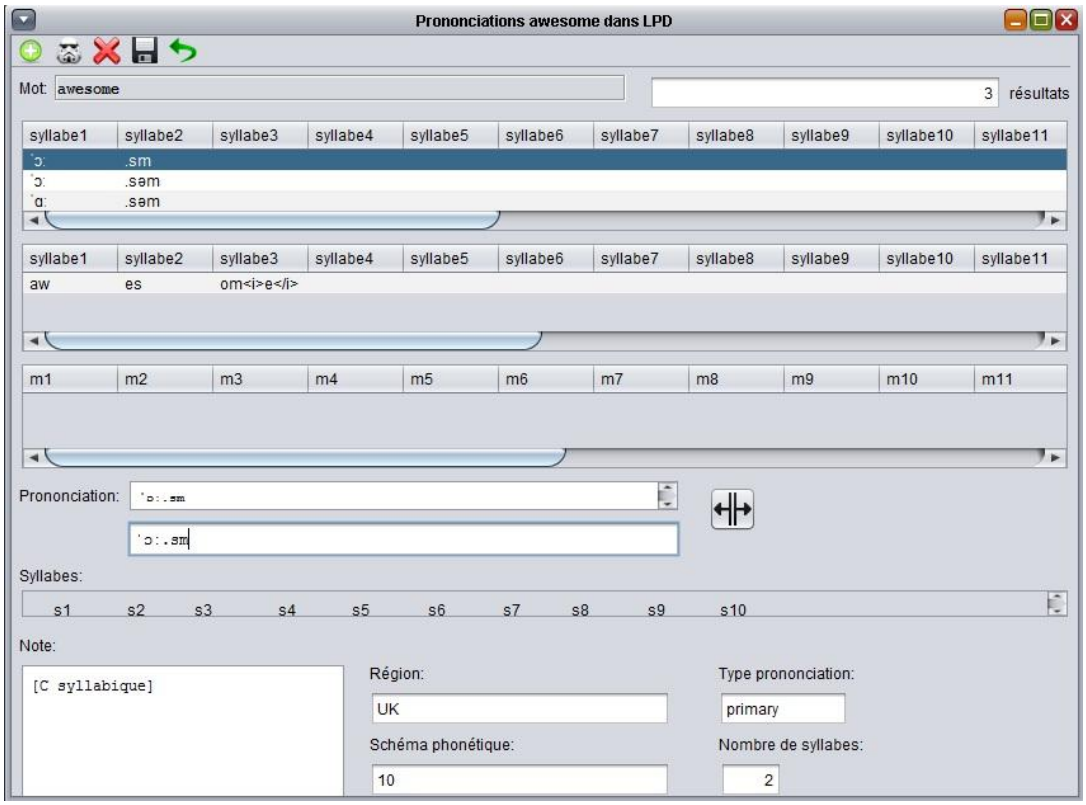


Figure 3 : Visualisation de la prononciation d'un mot

4.3. Interfaces logiciel/logiciel

Afin d'accéder à la base de données, le logiciel passera par XAMPP et son serveur Apache. Ce serveur se trouve sur un poste maître et les modifications que les utilisateurs veulent apporter devront être rentrées dans la base de données par l'utilisateur maître.

Voir le MCD de la base de données en Annexe de ce document.

5. Architecture générale du système

Le système est composé d'une base de données qui contient les mots des trois dictionnaires et les informations liées à ceux-ci. Cette dernière communique avec l'application via XAMPP / Apache.

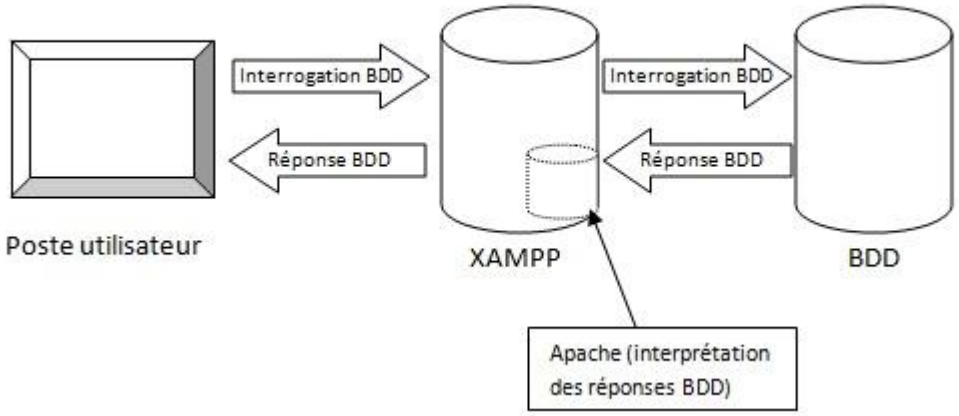


Figure 4 : Schéma des relations entre les différents éléments de l'architecture

Il existe deux types d'utilisateurs. Tous les utilisateurs auront accès en lecture à la base de données alors qu'un seul n'aura l'accès en écriture. Les autres utilisateurs feront leurs modifications en interne et transmettront ensuite un fichier contenant celles-ci à l'utilisateur maitre, qui enregistrera les changements dans la base.

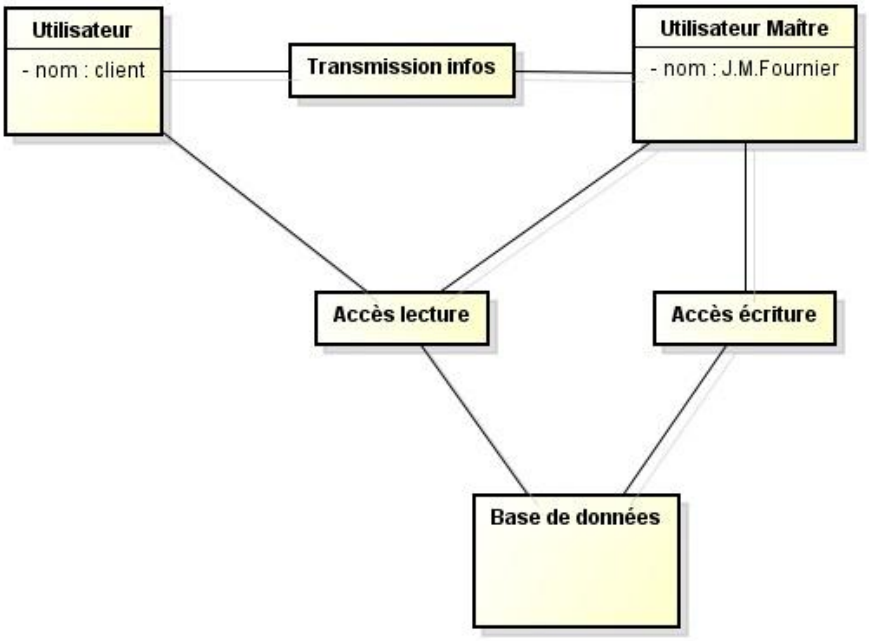


Figure 5 : Diagramme objet des droits utilisateurs

6. Description des fonctionnalités

6.1. Définition de la fonction de traitement de chaînes

6.1.1. Identification de la fonction de traitement de chaînes

Cette fonctionnalité est présente dans différentes tâches du projet :

- L'éclatement des mots en syllabes
- La construction du schéma phonologique des mots
- L'analyse des chaînes

Elle sera utilisée dans ces différentes tâches et permettra donc de traiter les chaînes et d'en tirer les informations nécessaires.

6.1.2. Description de la fonction de traitement de chaînes

Cette fonction prendra en entrée les fichiers XML contenant tous les mots des trois dictionnaires utilisés. Elle interagit avec la base de données, notamment pour l'analyse des chaînes où la demande de l'utilisateur est telle qu'il pourra interroger la base de données sur les syllabes des mots. Elle communique également avec la base de données pour l'enregistrement de ses résultats, comme les syllabes obtenues après éclatement du mot, le schéma phonologique ...

6.2. Définition de la fonction de traitement de la base de données

6.2.1. Identification de la fonction de traitement de la base de données

Cette fonctionnalité est très importante, car la base de données est le cœur du projet. Son but est de rajouter certaines informations à la base de données comme la fréquence d'utilisation des mots.

6.2.2. Description de la fonction de traitement de la base de données

Cette fonctionnalité sera entièrement consacrée à la base de données, et n'interagira qu'avec cette dernière.

6.3. Définition de la fonction de manipulation de fichiers

6.3.1. Identification de la fonction de manipulation de fichiers

Cette fonctionnalité est assez complète, puisqu'elle regroupe trois tâches :

- le parsage des fichiers XML
- l'enregistrement des modalités de recherche dans différents fichiers
- l'enregistrement des transferts d'ordres SQL au fur et à mesure des ordres, au lieu d'attendre la fermeture de l'application.

Elle permettra donc de parser les fichiers XML, d'enregistrer les modalités de recherche, et les transferts d'ordres SQL. Cette fonctionnalité interviendra à différents moments du projet, pour l'une où l'autre de ces tâches.

6.3.2. Description de la fonction de manipulation de fichiers

La fonctionnalité prendra comme entrée les fichiers XML contenant tous les mots des trois dictionnaires utilisés dans le cas du parsage XML. Pour ce qui est de l'enregistrement des modalités de recherche et des transferts d'ordres SQL, elle n'aura pas vraiment de paramètres d'entrée, mais elle gardera en mémoire les actions effectuées par l'utilisateur via l'interface.

Cette fonction interagira avec des fichiers, soit en lecture, soit en écriture, soit les deux, et interagira également, selon les actions de l'utilisateur, avec l'interface (pour synthétiser, elle interagira indirectement avec l'utilisateur via l'interface).

6.4. Définition de la fonction de traitement de l'interface

6.4.1. Identification de la fonction de traitement de l'interface

L'interface est un élément important car c'est elle qui interagit avec l'utilisateur. Toutes les commandes d'interrogation de la base de données sont données via cette interface.

Elle est déjà existante mais certains points sont à reprendre et à améliorer, selon l'évolution de la base de données, ou selon le souhait du client.

6.4.2. Description de la fonction de traitement de l'interface

La fonctionnalité traitera exclusivement de l'interface. Il faudra la remanier sur certains points, supprimer ou rajouter des éléments d'information. Durant tout le projet, cette fonctionnalité pourra évoluer selon les tests et/ou réunions effectués avec le client.

7. Conditions de fonctionnement

7.1. Performances

Du point de vue de l'utilisateur : l'application doit être assez rapide par rapport à l'interrogation et de mises à jour de la base de données. Elle se doit aussi d'être lors des traitements de fichiers (lecture, écriture).

Du point de vue de l'environnement : à chaque fois qu'un utilisateur veut appliquer une modification sur la base de données, cette dernière est enregistrée dans un fichier temporaire « .sql », qui sera transmit par la suite à l'utilisateur maître.

7.2. Capacités

Ce projet a une limite naturelle : la limite de stockage de la base de données PhpMyAdmin, qui peut varier selon les versions. La version utilisée actuellement est la 3.4.5.

7.3. Sécurité

Tous les utilisateurs du LLL ont accès à l'application. Cependant, la base de données en elle-même n'est stockée que sur l'ordinateur de M. FOURNIER ; comme expliqué précédemment, il ne fournit aux autres utilisateurs qu'une partie de la base de données, seulement la partie qui va être concernée par les modifications qu'ils vont effectuer dessus. Les utilisateurs transmettent ensuite un fichier contenant tous les ordres SQL effectués, et M. FOURNIER les exécute sur la base de données originale.

7.4. Intégrité

Comme je l'ai précisé plus haut dans ce document, les utilisateurs fournissent un fichier contenant tous les ordres SQL effectués sur la base de données lors de modifications de celle-ci. Actuellement ce fichier est édité automatiquement lorsque l'utilisateur quitte l'application. Cette dernière ne gère donc pas les cas de coupure inopinée, ce qui fait que l'utilisateur peut perdre toutes les commandes SQL qu'il a effectuées si cela venait à se produire.

M. FOURNIER m'a donc demandé de construire le fichier d'ordres SQL au fur et à mesure des commandes, et non à la sortie de l'application, pour éviter ce genre d'incident, et la perte complète du travail effectué.

7.5. Conformité aux standards

Pour toute la communication avec la base de données, le langage utilisé sera le langage SQL.
Le langage utilisé pour les fichiers issus des dictionnaires est le langage XML.

PLAN DE DEVELOPPEMENT

8. Découpage du projet en tâches

On indiquera également ici les tâches relatives à la gestion de projet (prise en mains de l'existant, bibliographie, rédaction du cahier de spécification, du rapport, de manuels techniques ou utilisateurs, mise en production et recette globale, etc.

8.1. Tâche 1 : Prise en main de l'existant

8.1.1. Description de la tâche

Cette tâche consiste en la compréhension de ce qui a déjà été fait à savoir :

- Le Modèle Conceptuel de Données
- Le parseur XML
- Le logiciel existant

8.1.2. Estimation de charge

L'estimation selon le diagramme de Gantt est de 10 Jours.

8.1.3. Contraintes temporelles

Il n'existe pas de contraintes temporelles fortes pour cette tâche.

8.2. Tâche 2 : Rédaction du cahier de spécifications

8.2.1. Description de la tâche

Cette tâche consiste en la rédaction du cahier de spécifications système.

8.2.2. Livrables

Le cahier doit être livré en fin de tâche. Il devra faire apparaitre les tâches à accomplir lors du projet et toutes les contraintes liées au projet.

8.2.3. Estimation de charge

L'estimation selon le diagramme de Gantt est de 13 Jours.

8.2.4. Contraintes temporelles

Le cahier de spécifications système doit être rendu avant le 30 Octobre 2011 pour sa première version et avant le 10 Novembre pour sa version finale sur l'intranet.

8.3. Tâche 3 : Refonte du parseur de fichiers XML

8.3.1. Description de la tâche

La base de données de l'application repose sur la fusion de trois fichiers XML, chacun de ces fichiers rassemblant les données de trois dictionnaires anglais :

- The Cambridge English Pronouncing Dictionary
- The Longman Pronunciation Dictionary
- The Macquarie Dictionary

Un parsage de ces fichiers est déjà existant mais il est fait pour l'ancienne structure de base de données. Il nécessite donc d'être modifié.

8.3.2. Cycle de vie

Différents tests sont prévus lors du développement de cette tâche, ainsi qu'à la fin.

8.3.3. Livrable

Cette tâche étant la première du projet, elle ne nécessite pas de livraison préalable d'élément pour être traitée.

8.3.4. Estimation de charge

L'estimation selon le diagramme de Gantt est de 8 jours si on ne compte que les jours réservés au PFE.

8.3.5. Contraintes temporelles

Cette tâche doit être réalisée avant les tâches concernant les outils d'application.

8.4. Tâche 4 : Outils de l'application

8.4.1. Description de la tâche

L'application qu'utilise actuellement le LLL comporte différents outils, dont deux qui sont assez importants :

- l'éclatement des mots en syllabes
- le schéma phonologique

Ces outils existent, mais ne fonctionnent pas vraiment. Il va donc falloir les reprendre, voir même les redévelopper complètement.

L'éclatement des mots en syllabes

Pour cette partie, il va falloir que les mots soient éclatés en syllabes, mais pas seulement en syllabes syntaxiques. Il faudra également éclater le terme phonétique correspondant au mot. Il faudra donc gérer ces deux sortes d'éclatement, qui ne vont pas se dérouler de la même façon, mais également gérer la correspondance entre ces deux termes éclatés.

Pour l'éclatement des termes phonétiques, il va également falloir gérer l'accentuation. Par exemple, une syllabe à accent principal sera précédée du signe « ' », une syllabe à accent secondaire du signe « ˌ », et une syllabe sans accent n'aura pas de signe. Cependant, le LLL a l'habitude de précéder les syllabes sans accent du signe « . ». Il faut donc gérer ces signes « . », les générer pour les syllabes sans signe particulier.

Le schéma phonologique

Le schéma phonologique se définit comme tel : « 1 »pour les syllabes à accent principal, « 2 » pour les syllabes secondaires, et enfin « 0 » pour les syllabes sans accent. Ce schéma s'appuiera bien entendu sur le point précédent.

8.4.2. Cycle de vie

Différents tests sont prévus lors du développement de cette tâche, ainsi qu'à la fin.

8.4.3. Livrable

Pour pouvoir être réalisée, il faut tout d'abord que le parsage des fichiers XML soit terminé.

8.4.4. Estimation de charge

L'estimation selon le diagramme de Gantt est de 8 jours si on ne compte que les jours réservés au PFE.

8.4.5. Contraintes temporelles

Cette tâche doit être réalisée avant la tâche d'analyse de chaînes.

8.5. Tâche 5 : Modification de l'interface

8.5.1. Description de la tâche

Dans l'application actuelle, quelques points de l'interface sont à revoir, comme des champs en trop pour l'éclatement des syllabes, des fenêtres d'interrogation/édition qui ne sont pas judicieusement placées, ...

8.5.2. Cycle de vie

Différents tests sont prévus lors du développement de cette tâche, ainsi qu'à la fin.

8.5.3. Livrable

Cette tâche n'a pas de contrainte de livrable dont elle dépendrait pour être réalisée.

8.5.4. Estimation de charge

L'estimation selon le diagramme de Gantt est de 6 jours si on ne compte que les jours réservés au PFE.

8.5.5. Contraintes temporelles

Cette tâche doit être réalisée avant la tâche des modalités de recherche.

8.6. Analyse des chaînes

8.6.1. Description de la tâche

Cet outil n'existe pas pour le moment. Il faudra donc le développer intégralement.
Pour comparer avec l'application courante, la recherche de mots peut se faire avec des spécificités précises : commence par, se finit par ...

Le nouvel outil à développer se comportera de la même façon, c'est-à-dire que l'utilisateur pourra spécifier s'il cherche un mot dont la première syllabe est accentuée, sans consonne, avec ou sans « e », etc... mais pourra en plus définir des variables qui seront enregistrées dans une base de variables.

Des outils effectuant ce genre de tâche doivent exister sur le marché, il suffira de les adapter aux besoins du LLL.

8.6.2. Cycle de vie

Cette tâche sera testée pendant et après la phase de développement

8.6.3. Livrable

Cette tâche nécessite que la tâche des outils de l'application soit d'abord terminée pour pouvoir être développée.

8.6.4. Estimation de charge

L'estimation selon le diagramme de Gantt est de 6 jours si on ne compte que les jours réservés au PFE.

8.7. Les modalités de recherche

8.7.1. Description de la tâche

Actuellement, l'application permet d'effectuer des recherches « en cascade » : l'utilisateur sélectionne d'abord un groupe de mot dans la base de données, puis il effectue une recherche sur ce groupe de mots, isole le résultat, effectue une recherche dessus et ainsi de suite.

Avec l'application courante, tous ces groupes de résultats sont regroupés sur un seul et même fichier, ce qui oblige à parcourir tout le fichier pour trouver le résultat, et ce qui est handicapant lorsqu'un chercheur du LLL veut fournir ses résultats à quelqu'un, mais sans lui fournir tous les groupes de recherche intermédiaires. M. Fournier voudrait que ces différents groupes de résultats soient placés dans des fichiers différents, pour n'avoir à la fin que les résultats finaux sur un fichier indépendant.

8.7.2. Cycle de vie

Cette segmentation des recherches en différents fichiers sera testée pendant et après la phase de développement.

8.7.3. Livrable

Basiquement, cette tâche n'a pas besoin de tâche précédente pour être développée, même si la répartition dans différents fichiers peut être faussée dans l'hypothèse où la base de données, ou l'éclatement des mots ou autres ne sont pas effectués.

8.7.4. Estimation de charge

L'estimation selon le diagramme de Gantt est de 4 jours si on ne compte que les jours réservés au PFE.

8.8. Enregistrement de la fréquence d'utilisation des mots

8.8.1. Description de la tâche

M. Fournier voudrait enregistrer la fréquence d'utilisation des mots, en collaboration avec une équipe de recherche américaine, qui a établi une base de données d'environ 400 millions de mots, avec leur fréquence d'utilisation.

8.8.2. Cycle de vie

Cette tâche n'étant pas une tâche de développement à proprement parlé, il n'y a pas de phases de tests nécessaires.

8.8.3. Estimation de charge

L'estimation selon le diagramme de Gantt est de 8 jours si on ne compte que les jours réservés au PFE.

8.8.4. Contraintes temporelles

Cette tâche n'a pas de contrainte de temps forte dans l'absolu, cependant on peut déjà penser à l'évolution du projet, et intégrer ces nouvelles informations dans la segmentation des recherches (Cf. modalités de recherche).

8.9. Transfert des ordres SQL client au serveur via un fichier

8.9.1. Description de la tâche

Actuellement, la structure informatique de l'application est la suivante : M. Fournier possède la base « serveur » sur son poste et il la distribue aux autres chercheurs lors de modifications. Dès qu'ils veulent modifier la base de données, les chercheurs qui sont donc clients de M. Fournier, lui transmettent un fichier contenant toute les modifications qu'ils souhaitent appliquer, que M. Founier approuvera ou non avec de les effectuer à sa BDD serveur.

Selon l'application actuelle, le ficher des ordres SQL client n'est généré qu'à la fermeture de l'application, ce qui peut être risqué en cas de coupure inopportune de l'application.
M. Fournier souhaiterait donc que le ficher soit généré au fur et à mesure, comme ça en cas de coupure de l'application, le travail effectué en amont n'est pas perdu.

8.9.2. Cycle de vie

Cette tâche sera testée pendant et après son développement.

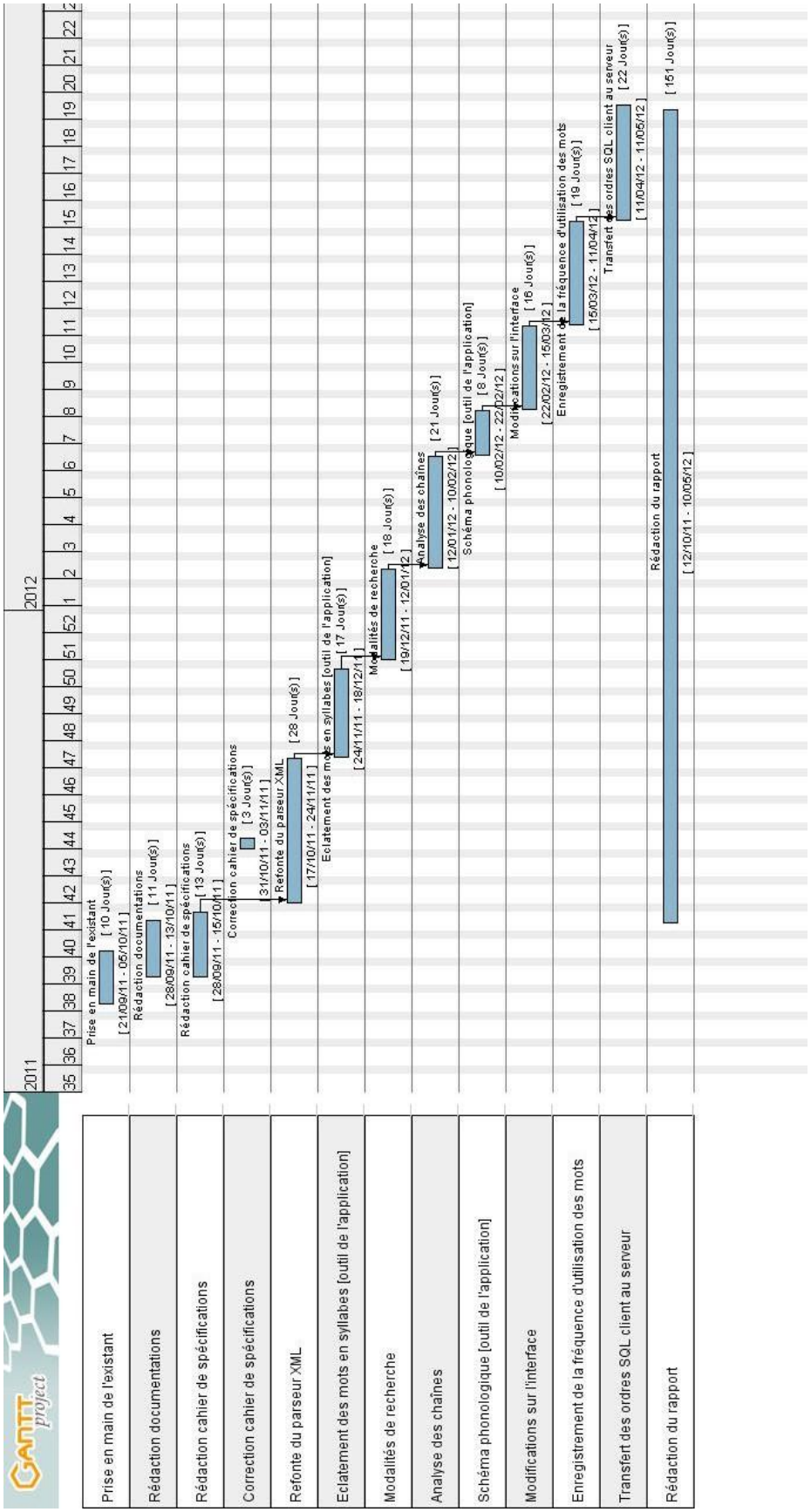
8.9.3. Estimation de charge

L'estimation selon le diagramme de Gantt est de 10 jours si on ne compte que les jours réservés au PFE.

8.9.4. Contraintes temporelles

Cette tâche étant la dernière selon le diagramme de Gantt, elle n'a pas de contrainte de temps forte mais doit être finie avant la fin du PFE.

9. Planning


GANTT project
2011
2012
35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
Prise en main de l'existant
[10 Jour(s)]
[21/09/11 - 05/10/11]
Rédaction documentations
[11 Jour(s)]
[28/09/11 - 13/10/11]
Rédaction cahier de spécifications
[13 Jour(s)]
[28/09/11 - 15/10/11]
Correction cahier de spécifications
[3 Jour(s)]
[31/10/11 - 03/11/11]
Refonte du parseur XML
[28 Jour(s)]
[17/10/11 - 24/11/11]
Eclatement des mots en syllabes [outil de l'application]
[17 Jour(s)]
[24/11/11 - 18/12/11]
Modalités de recherche
[18 Jour(s)]
[19/12/11 - 12/01/12]
Analyse des chaînes
[21 Jour(s)]
[12/01/12 - 10/02/12]
Schéma phonologique [outil de l'application]
[8 Jour(s)]
[10/02/12 - 22/02/12]
Modifications sur l'interface
[16 Jour(s)]
[22/02/12 - 15/03/12]
Enregistrement de la fréquence d'utilisation des mots
[19 Jour(s)]
[15/03/12 - 11/04/12]
Transfert des ordres SQL client au serveur
[22 Jour(s)]
[11/04/12 - 11/05/12]
Rédaction du rapport
[151 Jour(s)]
[12/10/11 - 10/05/12]
Prise en main de l'existant
Rédaction documentations
Rédaction cahier de spécifications
Correction cahier de spécifications
Refonte du parseur XML
Eclatement des mots en syllabes [outil de l'application]
Modalités de recherche
Analyse des chaînes
Schéma phonologique [outil de l'application]
Modifications sur l'interface
Enregistrement de la fréquence d'utilisation des mots
Transfert des ordres SQL client au serveur
Rédaction du rapport

GLOSSAIRE

LLL : Laboratoire Ligérien de Linguistique
SQL : Search Query Language
XML : eXtensible Markup Language

BIBLIOGRAPHIE

- Tous les documents liés à l'existant sont archivés
- Site internet sur la fréquence des mots : http://www.americancorpus.com