



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 spécifications système			
Projet :	Réalisation d'une base de données de dictionnaires linguistiques		
Emetteur :	BACCONNET Elodie	Coordonnées : EPU-DI	
Date d'émission :	18 novembre 2010		
Validation			
Nom	Date	Valide (O/N)	Commentaires

TACQUARD Claudine : 18/11/2010 ; O ;

Historique des modifications		
Version	Date	Description de la modification

01 : 03/11/2010 ; Version initiale

01.1 : 18/11/2010 ; Version finale - Précision des fonctionnalités

Table des matières

Cahier de spécification système	6
1.1 Introduction	6
1.2 Contexte de la réalisation	6
1.2.1 Contexte	6
1.2.2 Objectifs	6
1.2.3 Bases méthodologiques	6
1.3 Description générale	6
1.3.1 Environnement du projet	6
1.3.2 Caractéristiques des utilisateurs	7
1.3.3 Fonctionnalités et structure générale du système	7
1.3.4 Contraintes de développement, d'exploitation et de maintenance	7
1.4 Description des interfaces externes du logiciel	7
1.4.1 Interfaces matériel/logiciel	7
1.4.2 Interfaces homme/machine	7
1.4.3 Interfaces logiciel/logiciel	9
1.5 Architecture générale du système	10
1.6 Description des fonctionnalités	10
1.6.1 Définition de la fonction de traitement de chaînes	10
1.6.2 Définition de la fonction de traitement de la base de données	10
1.6.3 Définition de la fonction de manipulation de fichiers	11
1.6.4 Définition de la fonction de traitement de l'interface	11
1.7 Conditions de fonctionnement	11
1.7.1 Performances	11
1.7.2 Capacités	12
1.7.3 Sécurité	12
1.7.4 Intégrité	12
1.7.5 Conformité aux standards	12
Plan de développement	14
2.1 Découpage du projet en tâches	14
2.1.1 Parsage de fichiers XML	14
2.1.2 Restructuration de la base de données	14
2.1.3 Outils de l'application	15
2.1.4 Interface	16
2.1.5 Analyse des chaînes	16
2.1.6 Les modalités de recherche	17
2.1.7 Enregistrement de la fréquence d'utilisation des mots	17
2.1.8 Transfert des ordres SQL client au serveur via un fichier	18
2.2 Planning	18
A Glossaire	20

Cahier de spécification système

1.1 Introduction

Ce document présente les spécifications système du projet de fin d'études « Réalisation d'une base de données de dictionnaires linguistiques ». Il est en rapport avec le Laboratoire Ligérien Linguistique de l'université de Tours, que nous nommerons LLL dans la suite de ce document.

Actuellement, le LLL utilise une application existante, mais cette dernière ne correspond pas vraiment aux attentes du LLL. Mon travail consiste donc à reprendre ce projet, et l'améliorer, voire même refaire certains éléments.

1.2 Contexte de la réalisation

1.2.1 Contexte

Ce projet s'inscrit dans le cadre de la recherche linguistique, recherche effectuée par le LLL de Tours. Il concerne une application utilisée par les chercheurs, application qui regroupe toutes les données de trois dictionnaires anglais de prononciation :

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

Cette application permet aux chercheurs de répondre aux théories qu'ils expriment, et ce de façon simple et rapide, sans avoir à faire des recherches de façon indépendante sur ces trois dictionnaires. En effet, toutes les données sont accessibles via une seule et même application.

1.2.2 Objectifs

Le but de ce projet de fin d'études est de reprendre l'application actuelle du LLL, et de l'améliorer, de façon à ce qu'elle corresponde plus aux attentes des chercheurs.

Pour pouvoir réaliser cet objectif, le système s'appuiera sur PhpMyAdmin pour la base de données, et sur XAMPP pour l'exécution de l'application.

1.2.3 Bases méthodologiques

Ce projet sera réalisé en java pour la programmation, et la base de données sera une base de données MySql.

1.3 Description générale

1.3.1 Environnement du projet

Ce projet possède déjà un existant, existant qu'il faut reprendre et améliorer. La partie programmation sera effectuée en java, sous Netbeans 6.9, et la base de données MySql sera sous PhpMyAdmin. Tous ces outils seront synchronisés par XAMPP.

1.3.2 Caractéristiques des utilisateurs

Pour cette application, les utilisateurs seront les chercheurs du LLL. Cette application ne sera donc utilisée qu'en interne au LLL.

Les connaissances requises en informatique ne seront pas très importantes, les utilisateurs se servant déjà de l'application existante.

La future application reprendra dans l'ensemble les mêmes interfaces que l'application actuelle, ces dernières ayant juste besoin d'être modifiées/adaptées aux futures modifications apportées.

Tous les utilisateurs auront un même accès à l'application, mais seul l'utilisateur maître aura la base de données en totalité, les autres utilisateurs n'ayant accès qu'à une partie de cette base, lors de modifications.

1.3.3 Fonctionnalités et structure générale du système

Les fonctions utilisateurs du système seront la consultation de la base de données, la modification ou l'ajout d'informations sur un mot.

Le système est uniquement constitué de la base de données, et des interfaces permettant d'y accéder.

1.3.4 Contraintes de développement, d'exploitation et de maintenance

Contraintes de développement

Ce projet reprenant un projet déjà existant, les contraintes sont telles qui faut rester le plus proche possible du projet actuel, dans le sens où il faut reprendre la configuration déjà existante :

- * programmation en java
- * éditeur : Netbeans 6.9
- * base de données MySql
- * située sous PhpMyAdmin
- * kit d'installation Apache : XAMPP
- * fichiers sources des BDD en XML

1.4 Description des interfaces externes du logiciel

1.4.1 Interfaces matériel/logiciel

L'interface matériel/logiciel de ce projet n'est pas très conséquente : en effet, un simple ordinateur avec la configuration nécessaire d'installée (Apache, PhpMyAdmin, Netbeans, ...) et les données des dictionnaires peut exploiter l'application sans problème.

1.4.2 Interfaces homme/machine

Voici quelques exemples de l'interface actuelle :

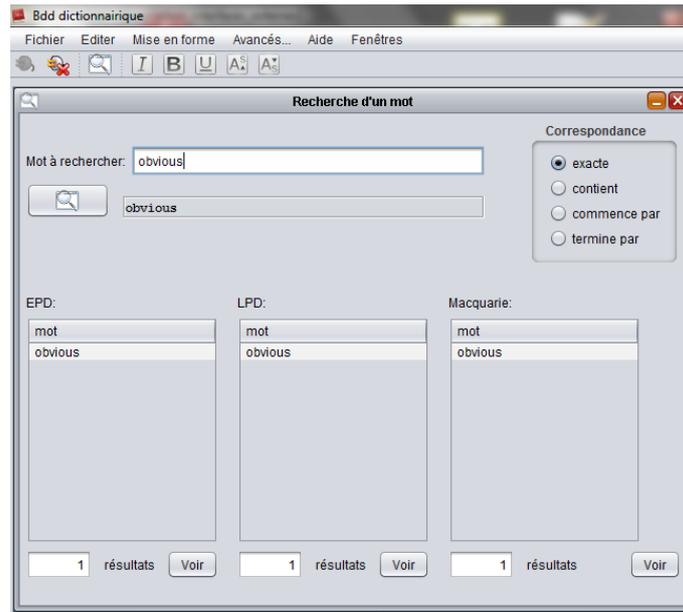


FIGURE 1.1 – Exemple d'interface : recherche de mot dans les dictionnaires

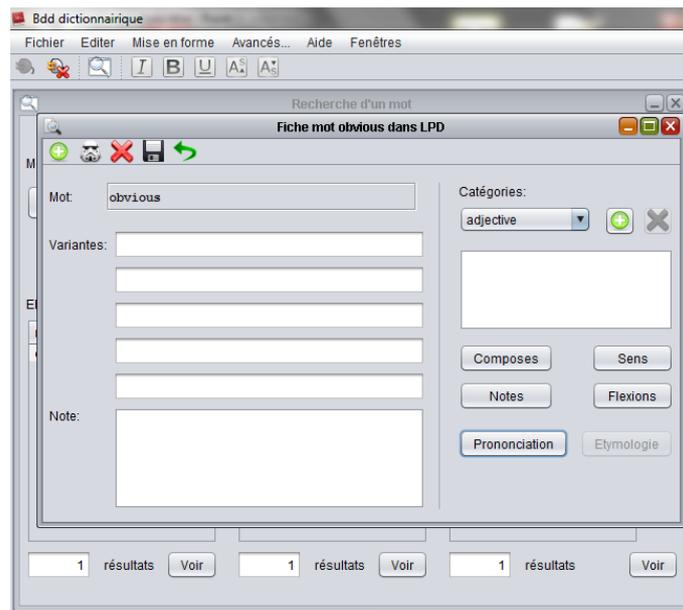


FIGURE 1.2 – Fiche issue du dictionnaire LPD du mot recherché

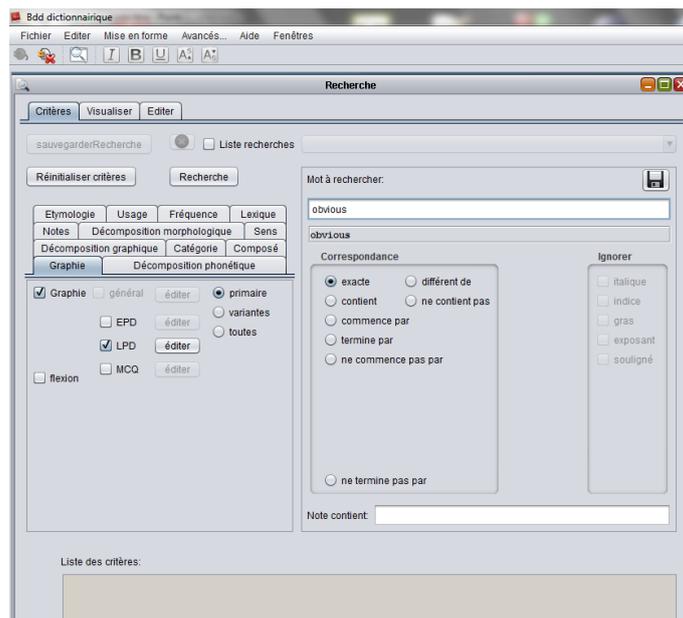


FIGURE 1.3 – Autre type de recherche : la recherche avancée

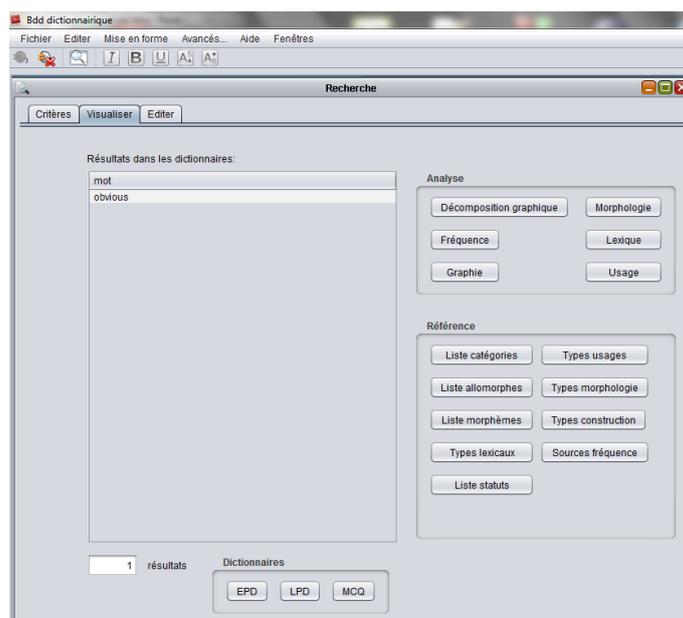


FIGURE 1.4 – Visualisation du mot recherché dans le dictionnaire LPD

Dans l'ensemble, l'interface ne sera pas modifiée. Seuls quelques détails seront revus.

1.4.3 Interfaces logiciel/logiciel

Pour pouvoir accéder à la base de données, l'application passera par le serveur Apache (XAMPP). La base de données sera centralisée sur le poste de l'utilisateur maître, et chaque utilisateur faisant des modifications à cette base de données devra passer par cet utilisateur maître pour que la modification effectuée soit effective pour tous les autres utilisateurs.

1.5 Architecture générale du système

Le projet a une architecture générale plutôt simple : en effet, le système est composé d'une base de données, qui contient tous les mots, et les informations relatives à ces derniers. Cette base de données communique avec l'application en elle-même via XAMPP / Apache.

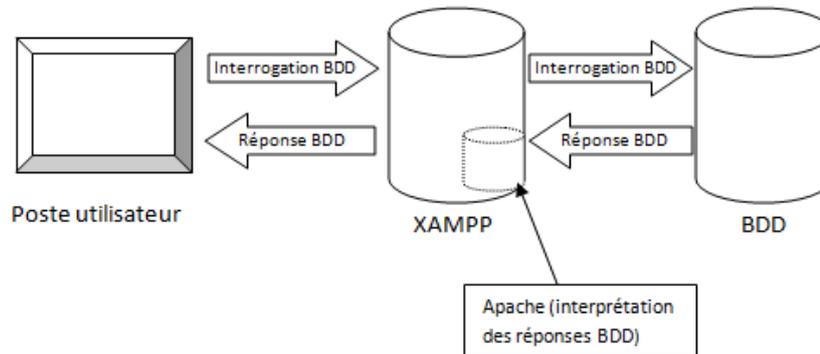


FIGURE 1.5 – Schéma des relations entre les différents éléments de l'architecture

1.6 Description des fonctionnalités

1.6.1 Définition de la fonction de traitement de chaînes

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

Cette fonctionnalité sera utilisée dans ces différentes tâches du projet, et permettra donc de traiter les chaînes et d'en tirer les informations nécessaires.

Description de la fonction de traitement de chaînes

La fonctionnalité prendra comme entrée les fichiers XML contenant tous les mots des trois dictionnaires utilisés.

Cette fonction 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 ...

1.6.2 Définition de la fonction de traitement de la base de données

Identification de la fonction de traitement de la base de données

Cette fonctionnalité est très importante, car la base de données est le coeur du projet.

Le but de cette fonctionnalité est de redéfinir la structure de la base de données, ainsi que d'y rajouter certaines informations, comme la fréquence d'utilisation des mots.

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.

1.6.3 Définition de la fonction de manipulation de fichiers

Identification de la fonction de manipulation de fichiers

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

- * le passage 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 une de ces tâches.

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 passage 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, en écriture, ou en lecture/écriture, et interagira également avec l'interface, selon les actions de l'utilisateur (pour synthétiser, elle interagira indirectement avec l'utilisateur via l'interface).

1.6.4 Définition de la fonction de traitement de l'interface

Identification de la fonction de traitement de l'interface

L'interface est un élément important, elle 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.

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.

1.7 Conditions de fonctionnement

1.7.1 Performances

Du point de vue de l'utilisateur, il est évident que l'application se doit d'être assez rapide au niveau de l'interrogation de la base de données, ainsi qu'assez efficace dans le traitement de fichiers (lecture, écriture, lecture/écriture).

1.7.2 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 3.2.4.

1.7.3 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.

1.7.4 Intégrité

Comme décrit précédemment, les utilisateurs fournissent un fichier contenant tous les ordres SQL effectués sur la base de données lors de modifications de cette dernière. Actuellement, ce fichier est édité automatiquement lorsque l'utilisateur quitte l'application.

Actuellement, l'application ne gère pas les cas de coupure inopinée, ce qui fait que l'utilisateur perd toutes les commandes SQL qu'il a effectuées en cas de coupure de l'application.

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.

1.7.5 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 développement

2.1 Découpage du projet en tâches

2.1.1 Parsage de fichiers XML

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

Le parsage de ces fichiers est déjà existant, seulement il est défaillant sur certains points, et nécessite donc d'être repris en partie.

Cycle de vie

Lors du développement de cette tâche, des tests sont prévus au cours du développement, ainsi qu'à la fin du développement.

Livrables

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

Estimation de charge

L'estimation selon le diagramme de Gantt est de 4,8 jours si on ne compte que les jours réservés au PFE.

Contraintes temporelles

Cette tâche doit être réalisée avant la tâche des outils d'application.

2.1.2 Restructuration de la base de données

Description de la tâche

La base de données de l'application existe également, mais certaines tables ne sont pas optimales : en effet, la base contient une table rassemblant les différents liens entre les trois dictionnaires sur de mêmes mots, et cette table n'est pas vraiment efficace, la création des liens n'est pas intuitive, et leur modification/suppression n'est pas réelle.

Mon travail consistera donc à reprendre cette table, à revoir la gestion de liens, mais il faudra également que je réfléchisse à propos de la table « pron », qui contient les prononciations des mots, et qui nécessite peut-être d'être éclatée, pour être mieux gérée ensuite, de façon plus claire.

Cycle de vie

Cette tâche n'étant pas une tâche de développement, elle n'a pas besoin de tests ou de recettes.

Livrables

Cette tâche ne requiert pas de livrable particulier pour être effectuée.

Estimation de charge

L'estimation selon le diagramme de Gantt est de 2 jours si on ne compte que les jours réservés au PFE.

Contraintes temporelles

Cette tâche doit être réalisée avant la tâche des outils d'application.

2.1.3 Outils de l'application

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 que je les reprenne, voire même que je les redéveloppe 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 que je gère l'accentuation. Par exemple, une syllabe à accent principal sera précédée du signe « ' », une syllabe à accent secondaire du signe « ´ », 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.

Cycle de vie

Lors du développement de cette tâche, des tests sont prévus au cours du développement, ainsi qu'à la fin du développement.

Livrables

Pour pouvoir être réalisée, il faut tout d'abord que la restructuration de la base de données soit terminée, ainsi que le passage des fichiers XML.

Estimation de charge

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



Contraintes temporelles

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

2.1.4 Interface

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, ...

Cycle de vie

Des phases de tests seront effectuées pendant et après la réalisation de la tâche.

Livrables

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

Estimation de charge

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

Contraintes temporelles

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

2.1.5 Analyse des chaînes

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 qu'il cherche un mot dont la première syllabe est accentuée, sans consonne, avec ou sans « e », ...

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

Cycle de vie

Cette tâche sera testée pendant et après la phase de développement.

Livrables

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

Estimation de charge

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

Contraintes temporelles

Cette tâche n'a pas de contrainte temporelle.

2.1.6 Les modalités de recherche

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.

Cycle de vie

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

Livrables

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.

Estimation de charge

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

Contraintes temporelles

Cette tâche n'a pas de contrainte temporelle forte.

2.1.7 Enregistrement de la fréquence d'utilisation des mots

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.

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.

Livrables

Pour que cette tâche soit réalisée, la tâche de restructuration de la base de données doit être terminée.



Estimation de charge

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

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).

2.1.8 Transfert des ordres SQL client au serveur via un fichier

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. Les chercheurs, qui sont donc clients de M. Fournier, lui transmettent ensuite un fichier contenant tous les ordres effectués pour modifier la base de données, que M. Fournier applique ensuite à sa base de données serveur.

Selon l'application actuelle, le fichier 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 fichier 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.

Cycle de vie

Cette tâche sera testée pendant et après son développement.

Livrables

Dans l'absolu, cette tâche n'a pas besoin que d'autres tâches soient terminées pour pouvoir la commencer, cependant, si la restructuration de la base de données n'est pas terminée ou effectuée, les ordres SQL placés dans le fichier ont de fortes chances d'être erronés.

Estimation de charge

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

Contraintes temporelles

Cette tâche étant la dernière selon le diagramme de Gantt, elle n'a pas de contrainte de temps forte.

2.2 Planning

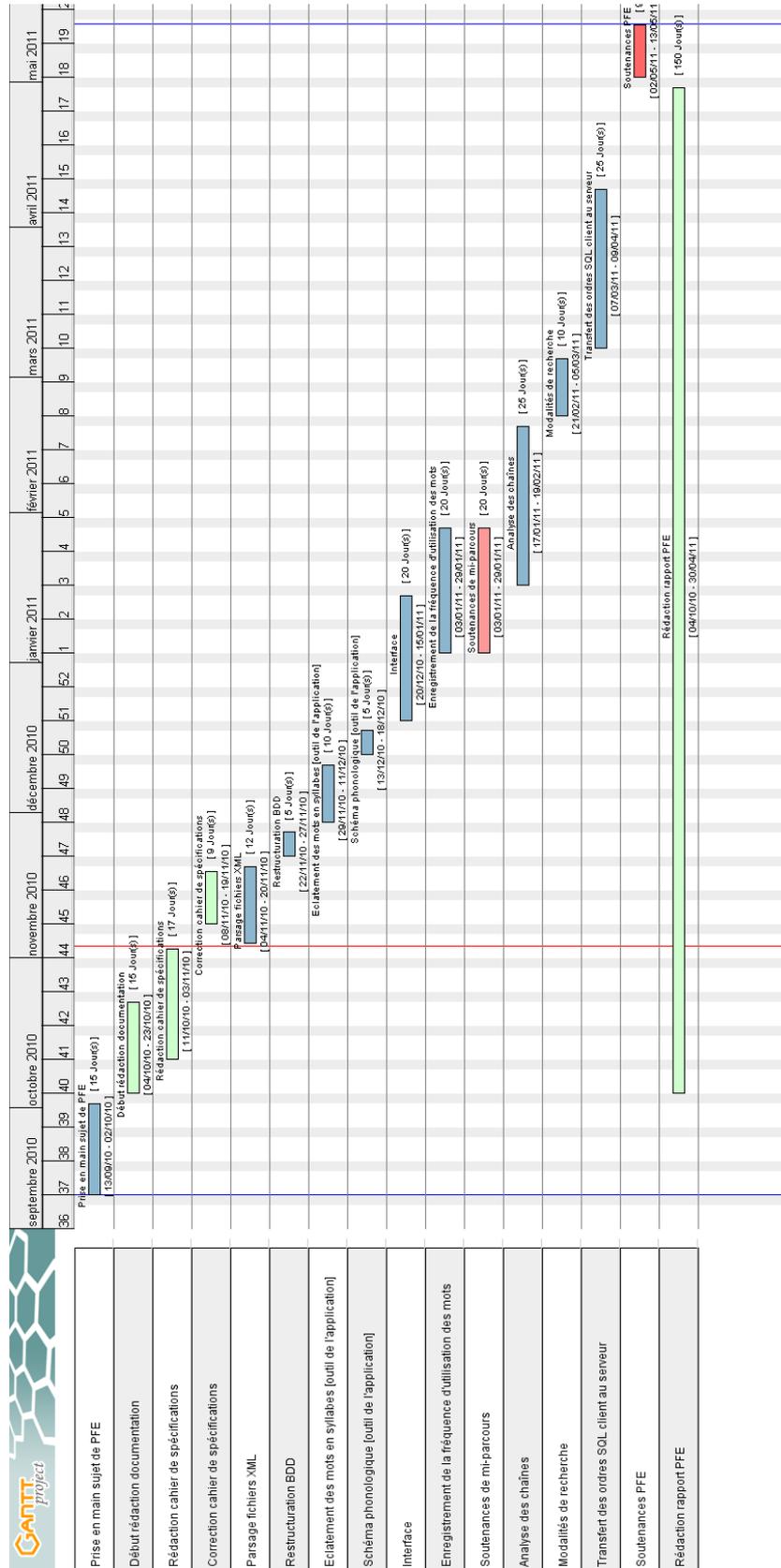


FIGURE 2.6 – Planning

Glossaire

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

Références

Site internet sur la fréquence des mots : [http ://www.americancorpus.org/](http://www.americancorpus.org/)