

Projet Xmlia

BOIVIN Benoit
LE PHILIPPE Noé
KEGBA-SANGO-SANGO Ulrich-Chancelin
WOUTERS Stéphane

22 mai 2014

Résumé

A remplir !

Remerciements

Nous remercions tout particulièrement Michel Meynard pour nous avoir brillamment encadré et soutenu tout le long de la réalisation de ce projet.

Table des matières

1	Introduction	3
2	Analyse du projet	4
2.1	Contexte	4
2.2	Analyse des besoins fonctionnels	4
2.3	Analyse des besoins non fonctionnels	5
2.3.1	Spécifications techniques	5
2.3.2	Contraintes ergonomiques	5
3	Rapport technique	6
3.1	Conception	6
3.2	Architecture de l'application	6
3.3	Résultat	6
4	Rapport d'activité	7
4.1	Organisation du travail	7
4.1.1	Communication	7
4.1.2	Répartition des tâches	7
4.2	Outils de développement	7
4.2.1	Langage et librairie	7
4.2.2	IDE	7
4.2.3	Gestionnaire de projet	7
5	Manuel d'utilisation	9
6	Perspectives et conclusions	10
6.1	Perspectives	10
6.2	Conclusions	10

Chapitre 1

Introduction

Chapitre 2

Analyse du projet

2.1 Contexte

De nos jours, le langage Extensible Markup Language ou XML est utilisé à des fins de stockage de données dont la forme est complexe (par exemple, un arbre). Ces données peuvent alors être organisées ou non selon un schéma qui va définir le format des données contenues dans le fichier XML.

Généralement, les fichiers XML sont soit générés par un programme quelconque, dans le but d'être lus par un autre programme, XML faisant office de plateforme commune. Mais le plus souvent ce sont des éditeurs de texte basiques qui sont utilisés pour créer de toute pièce un document XML, avec des fonctionnalités propre à un éditeur de texte, sans fonctionnalités spécialement prévues pour XML.

C'est dans ce contexte que des solutions logicielles d'éditeur XML ont vu le jour : un éditeur de texte qui possède des fonctionnalités permettant une écriture d'un fichier XML beaucoup plus rapide et efficace, le tout avec un contrôle des erreurs. Plusieurs solutions sont déjà proposées, certaines étant payantes (OxygenXML) et d'autres sont gratuites et open source (Xerlin). L'objectif est ici de fournir une solution similaire aux autres logiciels.

2.2 Analyse des besoins fonctionnels

L'objectif du projet est de développer un éditeur XML multi-vues avec différentes fonctionnalités.

Les fonctionnalités liées à un éditeur de texte simple devront être présentes : la possibilité de saisir manuellement au clavier l'intégralité du fichier, la création et la sauvegarde du fichier à manipuler ainsi que l'ouverture d'un fichier déjà existant dans le but de le modifier.

Des fonctionnalités d'éditeur de texte avancées seront aussi présentes : coloration syntaxique et indentation automatique du code permettant ainsi une lisibilité claire des fichiers manipulés et une autocomplétion du code écrit permettant un gain de temps au cours de la frappe.

Pour finir, l'éditeur proposera des fonctionnalités spécifiques au langage XML : validation syntaxique du fichier, vue arborescente du fichier XML avec possibilité de modification des données via cette vue et ajout d'un schéma sur lequel la validation se basera.

TODO : FAIRE UN SCHEMA D'INTERFACE

2.3 Analyse des besoins non fonctionnels

2.3.1 Spécifications techniques

Le programme devra permettre de créer des fichiers XML structurés avec un respect des normes de balisage et, s'il est défini, du schéma de données. De plus, les données saisies ou modifiées à l'aide de l'éditeur doivent rester exploitables, sans corruption possible du fichier original. Pour terminer, l'éditeur aura à répondre dans des durées acceptables et de manière stable, dans la mesure où la taille et la complexité des données restent raisonnables.

2.3.2 Contraintes ergonomiques

En termes d'ergonomie, l'utilisateur, dans la mesure où il connaît déjà le principe de XML et de son système de balisage, devra pouvoir prendre en main le logiciel en de brefs délais, avec un système d'icônes clair et des textes explicatifs, dans la mesure où il est en possession du manuel d'utilisation. De plus, un utilisateur avancé saura augmenter sa productivité en utilisant les raccourcis clavier disponibles, et ainsi gagner du temps au niveau de la transition souris/clavier, qui n'aurait donc plus lieu d'être.

Chapitre 3

Rapport technique

3.1 Conception

3.2 Architecture de l'application

3.3 Résultat

Chapitre 4

Rapport d'activité

4.1 Organisation du travail

4.1.1 Communication

La communication s'est au départ faite au travers de réunions hebdomadaires au cours desquelles le cahier des charges a été défini auprès de M. Meynard.

Une fois le cahier des charges défini et rendu, le but suivant a été de déterminer quel langage et quel bibliothèque d'affichage graphique sélectionner pour le projet et ainsi être conscient des avantages et des limites des éléments choisis.

Ensuite, l'objectif suivant a été de définir une maquette du logiciel afin de savoir quelles seront les fonctionnalités retenues, la manière de les exploiter et quel organisation visuelle du logiciel sera retenue. C'est ainsi qu'a été définie la maquette qui comporte une interface simple avec une barre d'outils, une vue principale sur le côté droit avec le système d'éditeur de texte, une vue secondaire sur la gauche avec la vue arborescente du modèle XML du fichier présenté par l'utilisateur et enfin la fenêtre de log associé au différents messages liés à l'utilisation de l'éditeur, par exemple, des erreurs de schéma.

Il fallait enfin définir l'organisation du travail au sein du groupe avec des tâches définies et mettre un place un plan de travail, c'est donc Stéphane WOUTERS, le chef du projet, qui a mis en place le gestionnaire de projet, a décidé des moyens de communication et qui a défini le premier objectif de développement : l'apprentissage de la librairie.

Puis la phase d'apprentissage et de développement commença, il fallait alors découvrir et apprendre le framework utilisé, et commencer à développer pour le projet, la communication s'est donc faite au travers de messages sur Google Hangouts, de mails, de commentaires via le gestionnaire de projet et de communication orale dans une salle informatique de la faculté.

4.1.2 Répartition des tâches

4.2 Outils de développement

4.2.1 Langage et librairie

4.2.2 IDE

4.2.3 Gestionnaire de projet

Le gestionnaire de projet utilisé est Bitbucket, un site Internet d'hébergement mutualisé supportant des projets utilisant Mercurial ou Git comme gestionnaire de versions. Dans le

cas de Xmlia, Git a été retenu et utilisé. (A FINIR)

Chapitre 5

Manuel d'utilisation

Chapitre 6

Perspectives et conclusions

6.1 Perspectives

6.2 Conclusions