

# **T.E.R.**

Moteur de présentation interactif guidé par la voix  
Master 1 Informatique

## Table des matières

|   |    |
|---|----|
| I.Remerciement.....                                     | 3  |
| II.Introduction.....                                    | 4  |
| III.État de l'art.....                                  | 5  |
| III.1.Interface vocale.....                             | 5  |
| III.2.Outils de présentation de diaporama.....          | 6  |
| IV.Projet existant .....                                | 7  |
| V. Organisation.....                                    | 8  |
| V.1.Outils fournis.....                                 | 8  |
| V.2.Travail prévu.....                                  | 8  |
| VI.Travail effectué.....                                | 8  |
| VI.1.Beamer/XML.....                                    | 9  |
| a)Choix technique .....                                 | 9  |
| b)Développement.....                                    | 9  |
| c)Gestion du temps.....                                 | 10 |
| VI.2.Client/Serveur.....                                | 10 |
| a)Choix techniques.....                                 | 10 |
| b)Développement.....                                    | 11 |
| c)Gestion du temps.....                                 | 12 |
| VI.3.Interface graphique.....                           | 12 |
| a)Choix technique.....                                  | 12 |
| b)Développement.....                                    | 12 |
| c)Gestion du temps.....                                 | 13 |
| VI.4.Clarification du code, documentation et tests..... | 13 |
| a)Gestion du temps.....                                 | 13 |
| VI.5.Diagrammes des outils.....                         | 14 |
| a)Non-existants (ce qui a été développé).....           | 14 |
| b)Existants.....  | 15 |
| VII.Difficultés rencontrées.....                        | 16 |
| VIII.Conclusion.....                                    | 16 |

## **I. Remerciement**

Nous tenons à remercier tout particulièrement notre tuteur, Monsieur Frédéric Béchet. Il a toujours été à notre écoute lors de nos différentes questions que cela soit par mail ou lors des différents rendez-vous. Il nous a aidé tout au long de ce T.E.R. Nous le remercions aussi de nous avoir permis d'utiliser les outils.

Nous remercions aussi Monsieur Benoît Favre, qui nous a aidé à installer les outils, et qui a répondu à toutes nos questions.

## II. Introduction

De nos jours, toutes les présentations se font à l'aide d'une souris, télécommande, clavier pour le contrôle des diapositives. Ce qui peut-être gênant pour un orateur lors d'une conférence.

Afin de remédier à tous ces problèmes, notre projet à pour but de faciliter l'interaction entre les diapositives et l'utilisateur à l'aide de la voix.

En se basant sur le discours énoncé, l'application pourra situer l'avancement de la présentation. Ainsi une fois que l'utilisateur finit d'exposer la diapositive courante, le programme passe à la suivante automatiquement.

La présentation devient alors plus naturelle par la fluidité du changement de chaque diapositive. En effet l'utilisateur devient libre de ses mouvements mais aussi de ses déplacements.

La méthode « Agile » a été utilisée pour la réalisation de ce projet. Cette méthode est incrémentale, c'est-à-dire que chaque semaine, un entretien avec le client a lieu. Ainsi le client peut voir l'état d'avancement du projet, mais aussi définir de nouveaux besoins.

Dans un premier temps, un logiciel de version a été mis en place pour le projet, afin que les membres de l'équipe puissent travailler et faire la fusion de leur travail plus facilement. Ensuite, un cahier des charges à été rédigé afin de cerner les besoins du projet.

La répartition des tâches fut facile étant donné le nombre restreint de personnes au sein de l'équipe ( 3 ).

Mlle Chikbouni s'est chargée principalement de la partie lecture d'un fichier Beamer, de la mise en place de la structuration et de la construction du fichier XML ainsi que de son écriture.

Mrs Lebreton et Rambaldi se sont occupés de la partie client/serveur, du moteur de diapositives ainsi que de la clarification et de la documentation du code.

Les membre du groupe se sont bien entendu retrouvés chaque semaine afin de procéder au bon développement du projet, mais aussi pour échanger des idées et s'entraider afin d'effectuer les différentes tâches dévolues à chacun.

## III. État de l'art

### III.1. Interface vocale

Depuis longtemps l'évolution de la Recherche et du Développement en matière informatique, l'augmentation de la puissance des ordinateurs et des réseaux, l'optimisation des lexiques et des phonèmes ont amené des changements radicaux sur l'efficacité et la fiabilité des systèmes vocaux.

Ainsi, depuis Windows Vista, le système d'exploitation de Microsoft intègre une fonction de reconnaissance vocale permettant aussi bien la dictée de textes que le pilotage de l'interface et des logiciels à la voix.

De la même manière, Microsoft a conçu en septembre 2008 un périphérique «Kinect» permettant de contrôler les jeux vidéos sans utiliser de manette. Elle permet d'interagir par commande vocale, reconnaissance de mouvement et d'image.

De la même façon, Apple a développé une application informatique de commande vocale qui comprend des instructions verbales données par les utilisateurs et qui répond à leurs requêtes. «Siri» permet de piloter à la voix les applications du système d'exploitation iOS telles le navigateur internet, l'envoi de messages, le téléphone ou encore la messagerie électronique.

#### Fonctionnement de la reconnaissance vocale

La reconnaissance vocale est une technique qui reconnaît dans une suite de signaux sonores les phonèmes et les phrases prononcées par un locuteur. Un son correspondant à un mot est prononcé dans un microphone qui convertit le signal acoustique en un signal électrique. Celui-ci est soumis à un prétraitement, puis subit la paramétrisation qui est une extraction de paramètres. Ce traitement s'applique tant en phase d'apprentissage qu'en phase de reconnaissance.

A partir d'un corpus d'apprentissage, trois types de modèle sont construits:

- Modèles acoustiques : consiste à superposer deux sons et à chercher le son qui se rapproche le plus de celui qui a été capturé par le microphone
- Modèles lexicaux : contient tous les mots à reconnaître avec toutes les prononciations possibles
- Modèles de langage : contient la probabilité d'apparition de chaque suite de mots

Ces trois types de modèle utilisés conjointement permettent de calculer la probabilité d'identifier toute suite de mots selon le signal vocal émis.

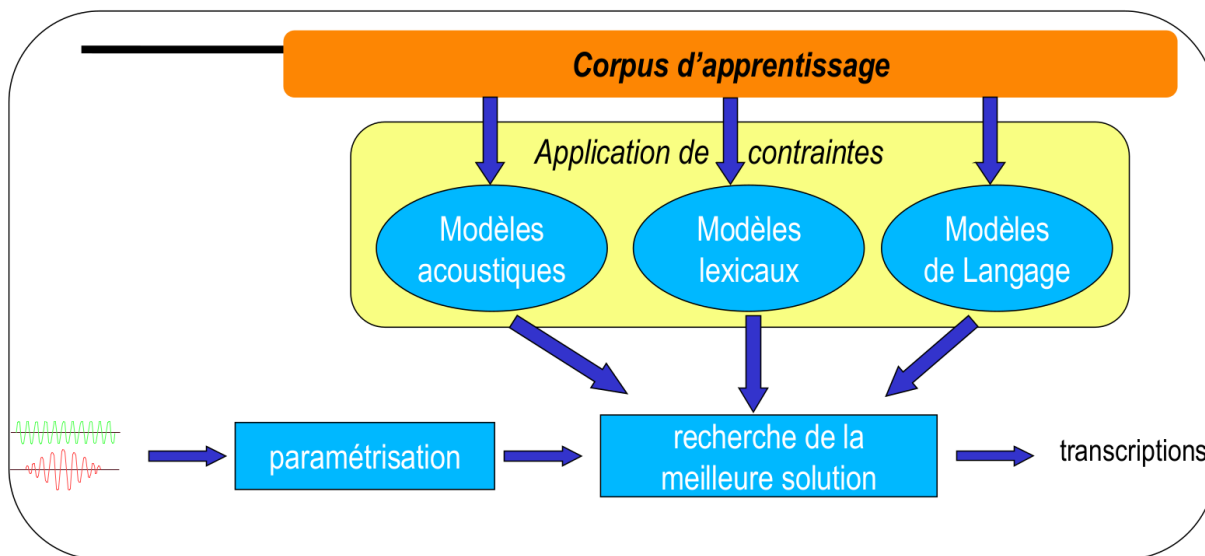


Illustration 1: Fonctionnement de la reconnaissance vocale

### III.2. Outils de présentation de diaporama

Il fut un temps où les conférenciers faisaient leur présentation à l'aide d'un tableau, ce qui n'était pas très pratique pour l'orateur.

Par la suite, l'utilisation des rétroprojecteurs rendit plus pratique toute présentation aussi bien pour le conférencier que pour l'auditoire. Cependant l'orateur était contraint de gérer les transparents, une manipulation qui prenait du temps et rendait la présentation moins dynamique.

Plus tard, les présentations numériques se firent jour, à l'aide de vidéo-projecteurs, d'ordinateurs et de logiciels de présentation tels PowerPoint ou Beamer. En fait de diapositives les premières présentations numériques n'utilisaient ni plus ni moins que des transparents améliorés ce qui les rendait statiques. Par la suite, PowerPoint a évolué en intégrant des animations, ce qui eut pour effet de rendre les présentations plus attractives pour tous.

Récemment, une alternative à PowerPoint nommée « Prezi » a été créée. C'est un outil de bureautique interactif qui permet de modéliser une présentation sans diapositives. L'interface est dotée d'un plan de travail virtuel, constitué de plusieurs espaces qui seront par la suite zoomés ou dé-zoomés. Les espaces de travail peuvent contenir du texte, des images, des vidéos et des liens.

## IV. Projet existant

Une version du moteur de présentation interactif guidé par la voix a été développée par Benoit Favre, Mickael Rouvier, Frédéric Bechet et Rocio Berenguer.

La version était adaptée pour un spectacle donné par des artistes espagnols s'exprimant en anglais. Comme il fallait prendre en compte un environnement bruyant et une musique de fond le logiciel de reconnaissance vocale devait offrir un niveau de performance élevé. De plus, l'accent des artistes était une contrainte supplémentaire à prendre en compte.

Dans cette version, l'outil « Kaldi » a été utilisé : il parvenait à récupérer les données en direct sans avoir recours à un enregistrement. Les paroles de l'artiste étaient d'abord retranscrites dans un fichier XML. Ce fichier était structuré par une liste de sections, chacune d'entre elles étant accessibles par un mot clé. Ces sections étaient composées de séquences comportant les paroles de l'artiste et des mots clés qui leur étaient associés.

Pour la communication entre l'artiste et le système, un protocole OSC était utilisé et le serveur associé déterminait l'action à faire à la réception d'un mot-clé.

Le projet de TER s'appuie sur ce précédent logiciel. Il fonctionne de la même manière, à l'aide d'un fichier XML qui décrit les actions à faire. Une liste de sections répertorie chacune des diapositives. Chaque section représentera une diapositive. Les données reçues par « Kaldi » seront envoyées via un protocole OSC au serveur qui effectuera l'action associée.

## V. Organisation

### V.1. Outils fournis

Plusieurs outils ont été fournis pour le développement de l'application.

- Kaldi : le logiciel de reconnaissance vocale.
- GSTKaldi : une version de « Kaldi » qui analyse le texte énoncé instantanément.
- Un outil qui permet de construire un modèle de langage à partir d'un fichier XML.

### V.2. Travail prévu

Afin d'atteindre l'objectif, trois phases ont été planifiées.

La première phase consistera à mettre en place une version qui fonctionnera avec deux mots clés. Le serveur s'occupera de passer à la diapositive suivante ou précédente selon le mot-clé reçu.

La deuxième version, permettra de changer la diapositive selon les mots clés qui leur correspond. Il sera aussi possible d'accéder à n'importe quelle diapositive.

La version finale consistera à changer de diapositive de façon automatique dès lors que le système détecte la fin du texte énoncé. Le système suivra donc l'évolution du texte, et changera la diapositive au moment opportun.

## VI. Travail effectué

Le développement de l'application se compose de trois parties distinctes :

- La partie Beamer/XML
- La partie Client/Serveur
- La partie Interface graphique



## VI.1. Beamer/XML

### a) Choix technique

Dans la partie concernant le fichier de présentation, le langage a été imposé. En effet, toutes les données utilisées pour la construction du fichier XML se trouvant dans le fichier beamer, il suffit de lire le fichier.

La bibliothèque JDOM est utilisée pour la partie concernant le XML . C'est une API (Application Programming Interface) permettant la manipulation de fichiers XML. Elle intègre notamment DOM (Document Object Model) qui est utilisé dans la lecture du document XML de ce projet, et SAX (Simple API for XML) qui est utilisé dans la création de document XML. L'utilisation de cette bibliothèque a été préconisée, car elle a été étudiée en cours d'XML.

### b) Développement

Pour récupérer les informations nécessaires au bon fonctionnement du programme, l'application commence par lire le fichier Beamer que l'utilisateur aura rédigé au préalable. Chaque information de chaque diapositive sera stockée. On pourra ainsi avoir accès aux commentaires, qui correspondent au texte que dira l'utilisateur, au texte contenu dans la diapositive mais aussi aux titres qui serviront éventuellement de mots clés.

Le fichier XML est formé de différentes balises :

- La balise « diapo » est la racine de ce fichier XML
- La balise « liste\_sections » représente le diaporama dans son ensemble. C'est-à-dire la liste de toutes les sections qui comprendront le texte énoncé par l'orateur pour chaque diapositives.
- Les balises keyword et keyword\_begin vont contenir les mot-clefs avec leurs actions en attribut.
- Les balises sections représentent une diapositive avec les mots-clefs qui lui correspondent ainsi que son texte.

La construction du fichier XML se fait avec les données récupérées du fichier Beamer. En effet en utilisant la bibliothèque JDOM (Java Document Object Model), le programme ajoute des nœuds contenant les commentaires du fichier Beamer, mais aussi les mots clés qui serviront à guider l'application pour les actions à faire.

Ainsi on pourra générer à chaque fichier Beamer un fichier XML qui lui correspond. Ce dernier comportera des mots-clés, et du texte qui sera prononcé par l'utilisateur.

Finalement, on aura un fichier comportant les balises nécessaires pour le contrôle des diapositives.

Pour la lecture du fichier XML, le Document Object Model (DOM) est utilisé. C'est un standard du W3C indépendant du langage utilisé. Cette API permet de modéliser, de parcourir et de manipuler un document XML. Le principal rôle de DOM est de fournir une représentation en mémoire d'un document XML sous la forme d'un arbre d'objets et d'en permettre la manipulation (parcours, recherche et mise à jour).

En ce qui concerne la lecture du fichier XML, il est lu en totalité, c'est-à-dire que sont lus toutes les balises, attributs et textes.

### c) Gestion du temps

| Tâche                    | Temps prévu | Temps consacré |
|--------------------------|-------------|----------------|
| Lecture fichier Beamer   | 5 jours     | 4 jours        |
| Construction fichier XML | 6 jours     | 7 jours        |
| Lecture du fichier XML   | 3 jours     | 5 jours        |

## VI.2. Client/Serveur

### a) Choix techniques

Dans cette partie, la bibliothèque JavaOSC (Java Open Source Control) est utilisée.

JavaOSC est une bibliothèque qui a été créée afin de faciliter la communication entre un client et un serveur avec le protocole OSC.

OSC est un protocole très utilisé pour la transmission de données vocales. Il est aussi conçu pour le contrôle en temps réel. Il utilise le réseau au travers des protocoles UDP ou TCP et apporte des améliorations en matière de rapidité et de flexibilité.

Le langage Java a été choisi car il est connu par tous les membres et ses nombreuses bibliothèques le rendent facile d'utilisation.

## b) Développement

Le client et le serveur communiquent selon le protocole OSC . Ayant reçu une démonstration du logiciel de reconnaissance vocale en Python, une première implémentation a été codée avec ce langage.

Cependant, la totalité du projet est développée en Java et un serveur a donc été codé en Java avec la bibliothèque JavaOSC. Cette API donne à Java la capacité d'envoyer et de recevoir des données via OSC.

Pour pouvoir intégrer les données de « Kaldi » au projet, la démonstration en Python a été transformée en client afin d'envoyer les voix détectées au serveur Java.

Le serveur joue un rôle important dans ce projet : analyser les données détectées par Kaldi afin d'effectuer des actions sur le visualiseur de diapositives. Par exemple changer de diapositive en fonction de certains mots-clefs énoncés par l'utilisateur ou bien détecter quand l'utilisateur arrive à la fin d'une diapositive pour décider de passer à la suivante.

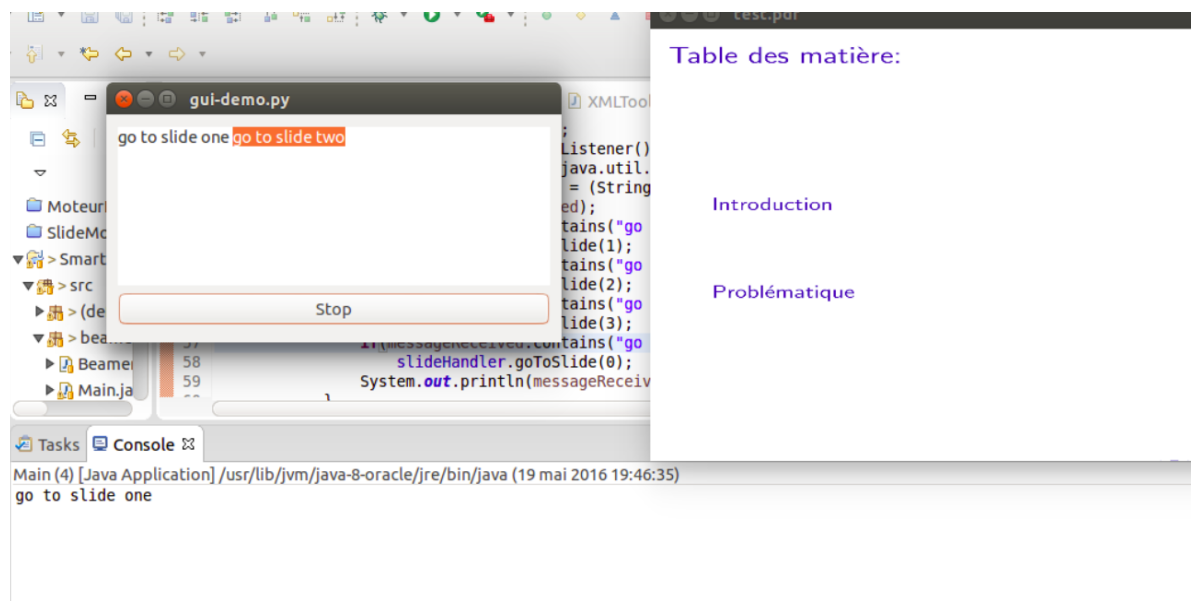


Illustration 3: Capture d'écran de la communication entre le client et le serveur

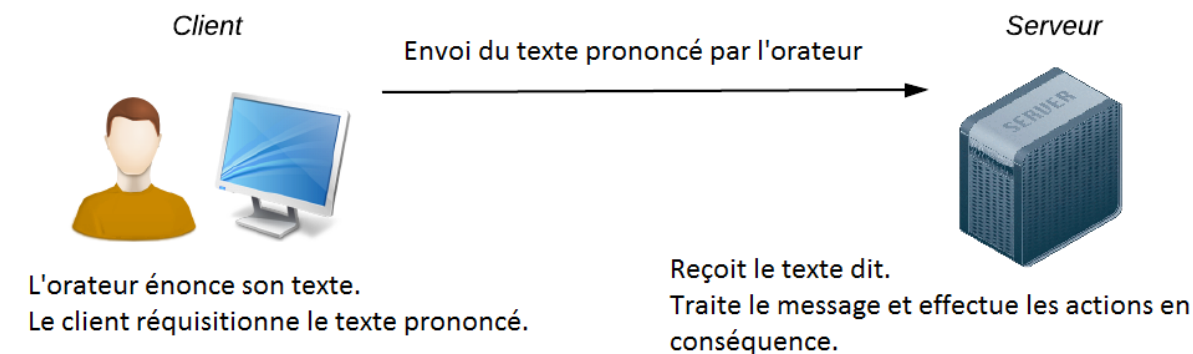


Illustration 4: Fonctionnement de la communication entre le client et le serveur

### c) Gestion du temps

| Tâche               | Temps prévu | Temps consacré |
|---------------------|-------------|----------------|
| Création du serveur | 5 jours     | 4 jours        |

## VI.3. Interface graphique

### a) Choix technique

L'interface graphique a été développée en premier lieu avec Swing, une bibliothèque graphique pour le langage Java. Elle a ensuite subi une refonte complète en raison du choix d'une nouvelle API graphique : JavaFX. Le développement d'applications avec Swing n'est plus d'actualité alors que la bibliothèque JavaFx est très populaire car c'est une API simple en ce qui concerne la compréhension et le développement, mais elle fournit aussi plus de possibilités: média audio, vidéo, graphisme 2D, 3D, etc.

PdfRenderer est une bibliothèque développée en java qui permet de manipuler les fichiers Pdf afin d'en récupérer toutes les informations comme du texte ou bien une page entière.

### b) Développement

Pour la partie affichage, la bibliothèque Java « PdfRenderer » a été utilisée afin de pouvoir découper un fichier PDF en plusieurs images. Ces images sont ensuite incluses dans une structure de données appelée « Slide » qui représentera en quelque sorte une page du fichier pdf avec les commentaires du fichier beamer associés à celle-ci. Chaque « Slide » représente une diapositive. Le changement de diapositive se fait de plusieurs façons : en prononçant le mot clé adéquat, en prononçant le texte entier de la diapositive ou bien via le clavier.

### c) Gestion du temps

| Tâche                               | Temps prévu | Temps consacré |
|-------------------------------------|-------------|----------------|
| Afficher des slides à partir du pdf | 1 jour      | 1 jour         |
| Pouvoir changer de diapositive      | 4 jours     | 5 jours        |
| Création de l'interface graphique   | 2 jours     | 3 jours        |

## VI.4. Clarification du code, documentation et tests

Tout le code du projet a été clarifié pour être plus lisible et accessible et une documentation a été écrite. De plus des tests ont été effectués tout le long du projet. A la fin du développement, des tests finaux ont été faits après la fusion des différentes parties.

### a) Gestion du temps

| Tâche                 | Temps prévu | Temps consacré |
|-----------------------|-------------|----------------|
| Clarification du code | 4 jours     | 2 jours        |
| Documentation         | 4 jours     | 2 jours        |
| Tests                 | 5 jours     | 7 jours        |

## VI.5. Diagrammes des outils

### a) Non-existants (ce qui a été développé)



Illustration 5: diagramme de classe de tous les outils développées au cours du projet

## b) Existants

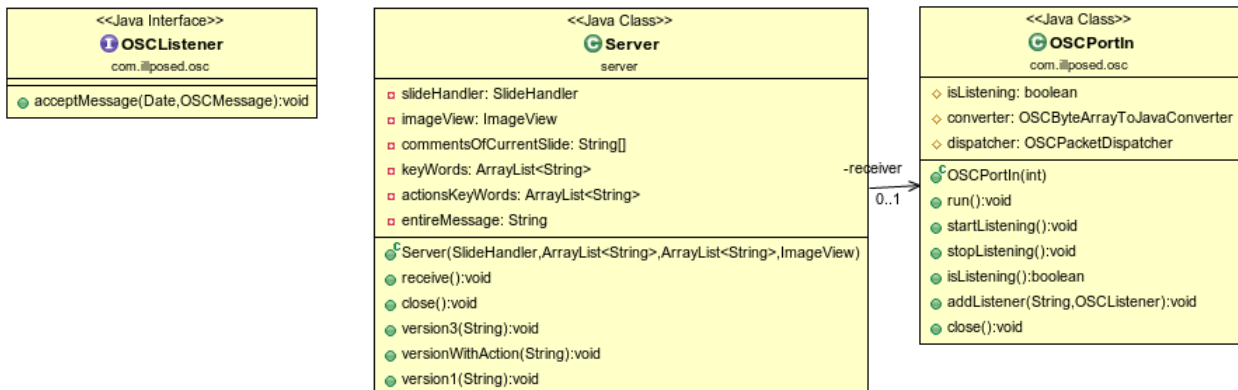


Illustration 6: Diagramme de classe du serveur et des outils existants (JavaOSC)



Illustration 7: Diagramme de classe du diaporama avec les composants existants(PdfRenderer)

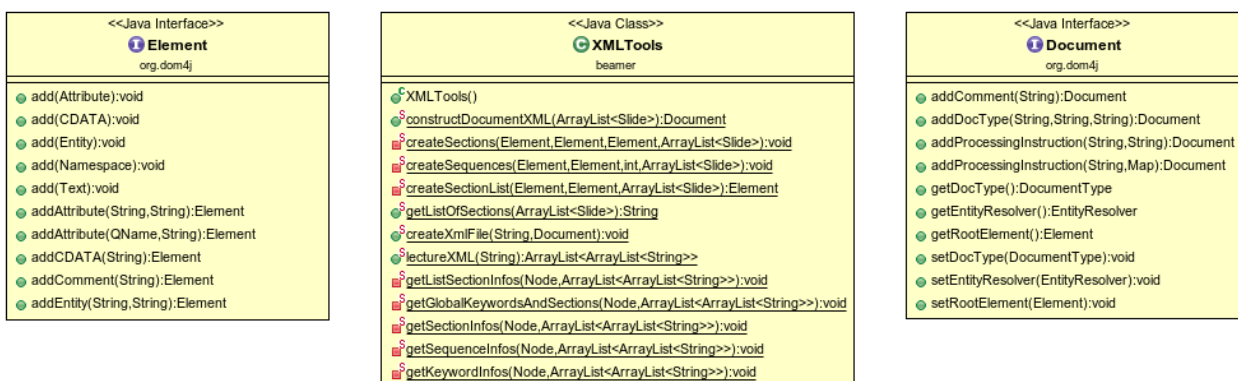


Illustration 8: Diagramme de classe des outils XML et de JDOM4

## VII. Difficultés rencontrées

Au départ, la communication entre le client en Python et le serveur en Java était difficile. Pour y remédier, le serveur a été codé en Python, afin d'en connaître le fonctionnement. Par la suite il a été plus facile de l'adapter en Java.

Une autre difficulté rencontrée fut que le logiciel de reconnaissance vocale n'arrivait pas à détecter parfaitement le discours énoncé par l'utilisateur. En effet les phrases qu'il affichait sur l'interface de démonstration ne correspondaient pas tout à fait aux phrases qui étaient dites. Cela influait directement sur les performances de l'application.

Grâce à l'aide de M. Favre lors de la dernière semaine, une nouvelle configuration des outils a été faite. Dès lors, le reconnaiseur vocal fonctionnait correctement.

## VIII. Conclusion

Finalement, l'application est opérationnelle, le changement de diapositives s'opère de manière orale et permet donc de faire des présentations interactives guidées par la voix. Le développement des actions permettant le changement de diapositive sans mots-clefs est presque fonctionnel, quelques points restant à améliorer pour que celui-ci marche parfaitement. L'utilisateur doit par contre être un connaisseur de LaTeX pour utiliser cette application.

Ainsi, on peut penser à plusieurs extensions de cette application. Comme par exemple, un conte lu par des enfants : au fur et à mesure de l'avancement de la lecture, les illustrations du conte défileraient. On peut également penser à une amélioration susceptible de personnaliser l'application de telle manière que cela ressemblera à un dialogue entre l'utilisateur et la machine. On peut aussi penser à une version qui permettrait de lire différents formats de diaporama, comme par exemple : PowerPoint, OpenOffice, Keynote.