

Mazigh Mouanes 1721035

Paul Clas 1946912

LOG2410 - TP2 : Diagramme de concepts et assignation des responsabilités

2. Diagramme du Cas d'Utilisation n°4 :

Cas d'utilisation : Traduction automatique d'un flux audio

Acteur : Utilisateur

Type : Primaire

Partie prenante et intérêts : Utilisateur désire se faire comprendre par un interlocuteur qui ne parle pas sa langue source. Interlocuteur doit comprendre le message traduit. La traduction doit être correcte et la voix synthétique compréhensible.

Description : Traduire automatiquement un flux audio d'une voix humaine s'exprimant dans une langue spécifiée (langue source) vers une autre langue (langue cible) avec une voix (voix synthétique) choisie.

Préconditions : L'utilisateur a l'application PolyVersion sur son cellulaire. L'utilisateur a téléchargé les dictionnaires des langues sources et cible pour la traduction. Le cellulaire possède un microphone et haut-parleurs.

Garanties en cas de succès : PolyVersion traduit et transmet via haut-parleur un message traduit et compréhensible par l'interlocuteur.

Scénario Principal :

1. Le client ouvre l'application PolyVersion.
2. Présentation des options.
3. L'utilisateur choisi de traduire un flux audio automatiquement.
4. Présentation des langues disponibles de traduction.
5. L'utilisateur choisi la langueCible.
6. Présentation des choix de medium de sorties.
7. L'utilisateur choisi les Haut-Parleurs.
8. Présentation du délai de traduction pour la transmission audio
9. L'utilisateur choisi le délai de 1,5 secondes.
10. Présentation des choix de voix synthétiques.
11. L'utilisateur choisi la voix synthétique.
12. PolyVersion informe l'utilisateur que PolyVersion est prêt pour la traduction Automatique (bouton vert à l'écran)
13. L'utilisateur commence à parler.
14. PolyVersion capture le flux audio grâce au microphone (non visible par l'utilisateur)
15. PolyVersion détecte la langueSource. (non visible par l'utilisateur)
16. PolyVersion traduit le flux audio de la langueSource à la langueCible (non visible par l'utilisateur)

17. PolyVersion émet le flux audio avec la voix synthétique choisi aux haut-parleurs. (non visible par l'utilisateur)
18. L'interlocuteur entend le flux audio traduit.
19. L'utilisateur arrête de parler.
20. PolyVersion arrête de transmettre après 1,5 secondes de l'arrêt du flux audio. (non visible par l'utilisateur)
21. L'interlocuteur a compris la traduction.
22. L'utilisateur ferme PolyVersion.

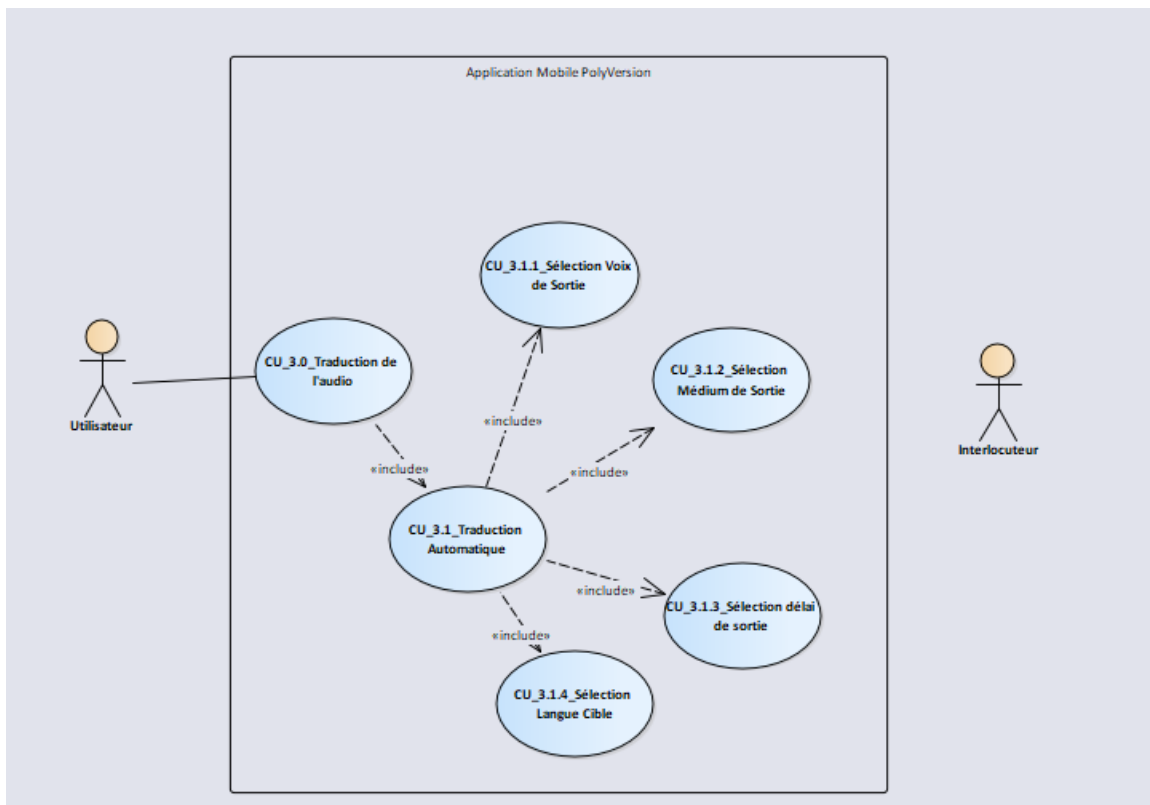


Figure 1 : Diagramme du cas d'utilisation 4 de PolyVersion

Nous avons réalisé un nouveau cas d'utilisation 4.0 pour réaliser ce travail car nous n'étions pas satisfait de nos cas d'utilisation présenté au tp1.

3. Question :

Expliquez-en vos propres mots la/les différence(s) entre le diagramme de concepts et le diagramme de classes :

Le diagramme de concept fait la construction d'un modèle simple qui révèle les principaux concept et relation d'un domaine de solution logicielle alors que le diagramme des classes indique l'application logicielle des concepts avec méthodes, responsabilités et fonctions.

Un diagramme de concept précède le diagramme des classes par son niveau d'abstraction.

Le diagramme de concept favorise la compréhension et constitue la base sur laquelle on peut fonder les classes d'un problème orientée-objet.

« Le diagramme de concepts est représenté par un diagramme de classes où les concepts sont présentés par des classes. Les classes conceptuelles sont en relation les unes avec les autres afin de former une description visuelle du domaine d'application. » - TP2, Log2410

« Un diagramme de concepts n'est pas une représentation des entités logicielles d'un système (concepts != classes logicielles). » TP2, Log2410

4. Assignment de responsabilité et patrons GRASP

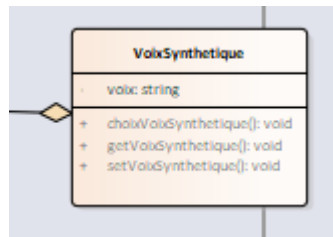


Figure 2 : Patron Expert de la Classe VoixSynthétique

VoixSynthétiques utilise le patron d'assignation expert pour se donner la responsabilité de déterminer et changer la voix synthétique pour la lecture du fichier car la classe et le mieux en mesure d'y satisfaire.

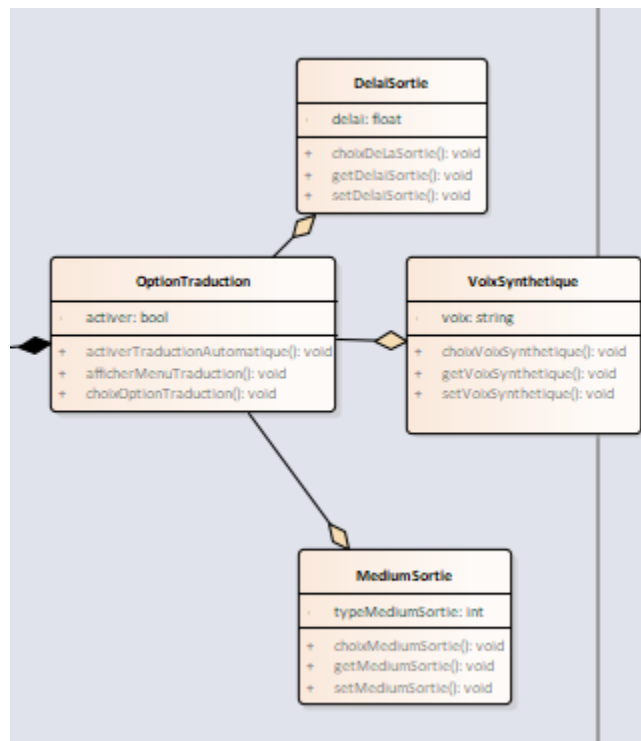


Figure 3 : Patron Créateur de la classe OptionTraduction

OptionTraduction utilise le patron d'assignation créateur car elle crée des classes DelaiSortie, ChoixMedium et VoixSynthetique pour chaque traduction dépendamment du choix d'option de traduction de l'utilisateur. OptionTraduction a la responsabilité de gérer les choix de la traduction.

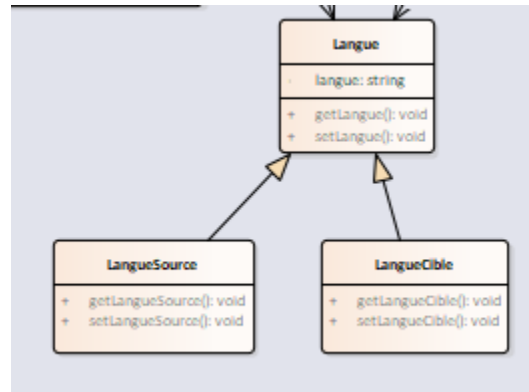


Figure 4 : Patron Polymorphisme de la classe Langue

Langue utilise le patron d'assignation de polymorphisme car elle crée deux classes qui dépendent des alternatives soit source ou cible. Langue a la responsabilité de gérer ces classes pour faciliter la traduction et la transmission d'attribut privé.

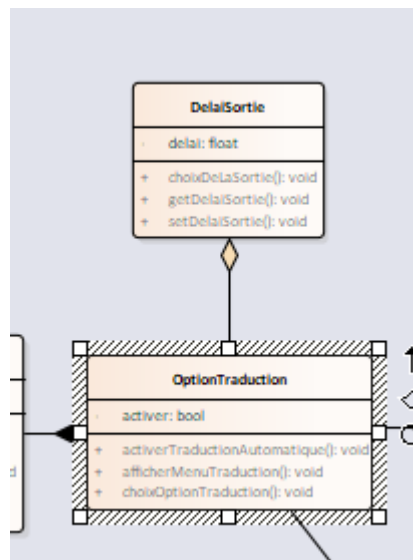


Figure 5 : Patron d'assignation de couplage faible de OptionTraduction et DelaiSortie

DelaiSortie utilise le patron d'assignation de couplage faible en étant couplé uniquement avec OptionTraduction ce qui rend DelaiSortie plus indépendante et améliore les gains en productivité. DelaiSortie aurait pu être couplé avec traduction ou même FluxDeSortie mais le couplage faible favorise l'efficacité. DelaiSortie a alors la seule responsabilité de déterminer et donner le délai de sortie désiré par l'utilisateur.

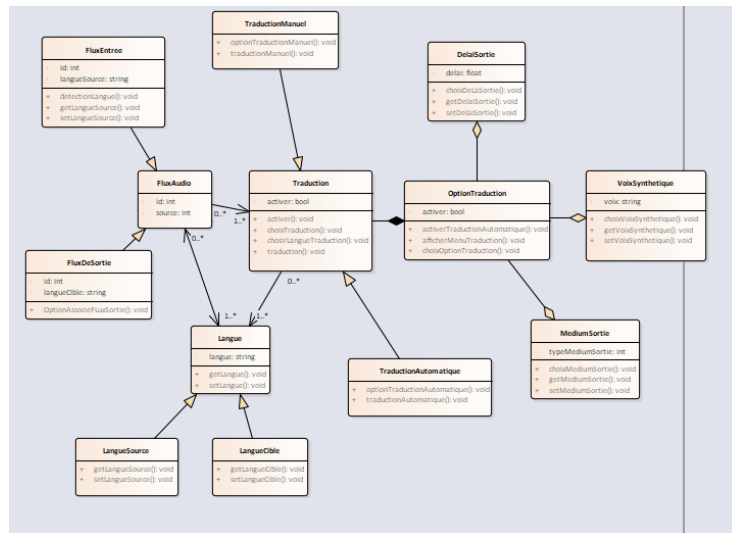


Figure 6 : Diagramme de classe de notre cas d'utilisation

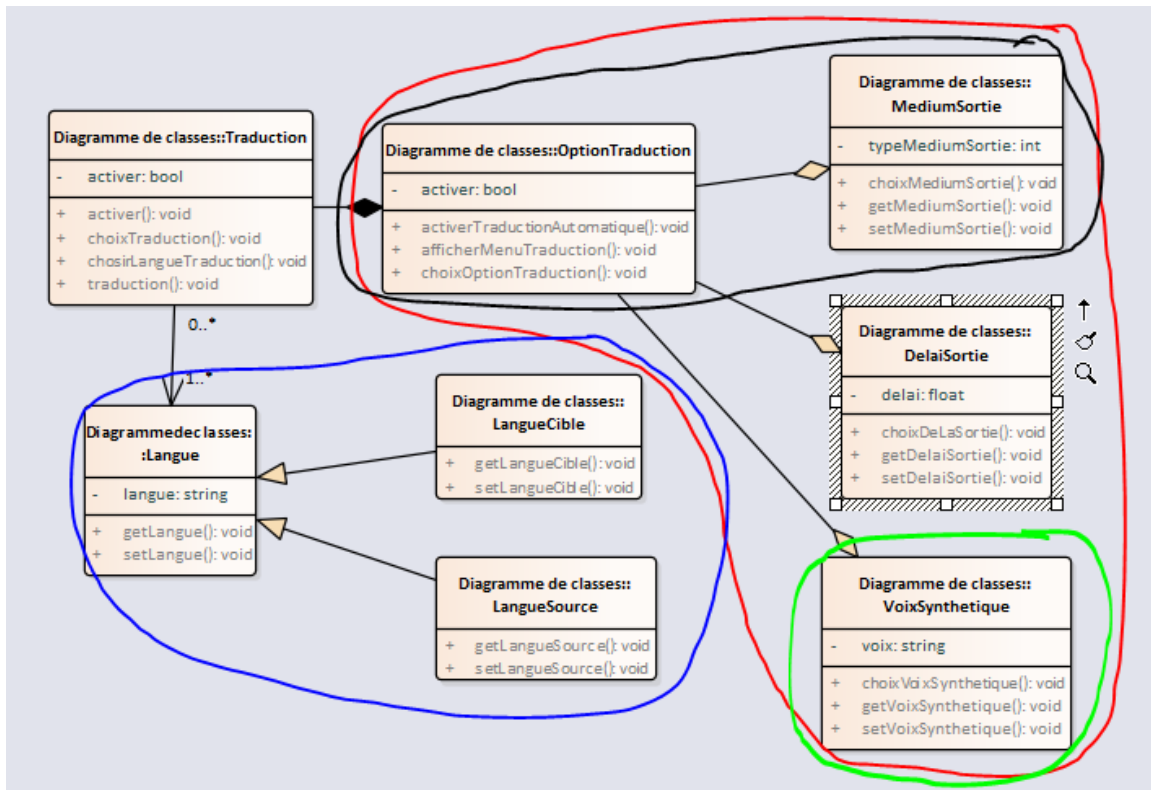


Figure 7 : Diagramme de Classe avec les patrons GRASP explicitement montré (en vert : Expert, en rouge : Créateur, en noir : Couplage Faible, en bleu : Polymorphisme)