



Plate-forme iMAG
DOSSIER DE CONCEPTION GLOBALE

Version 1.1

7 . 2011

Table des matières

| | | |
|----------|---|----------|
| 1 | Tableau de suivi des modifications | 2 |
| 2 | Introduction | 3 |
| 2.1 | But du document | 3 |
| 2.2 | Organisation du document | 3 |
| 3 | Prise en compte des facteurs qualité attendus | 5 |
| 3.1 | Rappel des facteurs attendus | 5 |
| 3.2 | Application des tactiques | 5 |
| 3.3 | Utilisabilité | 6 |
| 3.4 | Maintenabilité | 6 |
| 3.4.1 | Localiser les modifications | 6 |
| 3.4.2 | Eviter les effets de bord | 6 |
| 3.5 | Fiabilité | 6 |
| 4 | Le framework XWiki | 7 |
| 5 | Architecture | 9 |
| 5.1 | Vue logique | 9 |
| 5.1.1 | Vue logique pour version 1 | 9 |
| 5.1.2 | Vue logique pour version 2 | 10 |
| 5.1.3 | Description des composants | 10 |
| 5.2 | Vue physique | 10 |
| 5.3 | Vue dynamique | 11 |
| 5.3.1 | S'inscrire | 12 |
| 5.3.2 | Changer le code secret | 13 |
| 5.3.3 | Ajouter un corpus | 14 |
| 5.3.4 | Post-éditer | 15 |
| 5.3.5 | Utiliser dictionnaire | 16 |
| 5.3.6 | Créer et initialiser un site en plusieurs langues | 17 |

Chapitre 1

Tableau de suivi des modifications

| Version | Date | Per- sonne+Contact Info | Description |
|---------|----------|-------------------------------|-----------------------|
| 1.0 | 21/05/11 | Zhang Ying, Wang Lingxiao | Initialisation |
| 1.1 | 30/05/11 | Zhang Ying, Wang Lingxiao | Traitement de page |

Chapitre 2

Introduction

2.1 But du document

Le présent document a pour objectif de définir l'architecture logicielle de l'application SECTra_w et iMAG. Cette architecture repose sur la spécification des besoins donnée dans le Cahier Des Charges [CDC] et sur les spécifications externes données dans le Dossier de Spécifications Externes [DSE]. Ce document servira de base à la conception et au développement du produit à livrer. Les aspects suivants sont traités :

- Vue logique : Représentation de tous les composants de l'application et de leurs interactions.
- Vue physique : Cette vue montre comment les éléments de la vue logique sont alloués à des plateformes d'exécution.
- Vue dynamique : Vue permettant de modéliser la succession des activités et la nature des interactions entre les différents modules.

Les langages et frameworks de programmation utilisés sont également spécifiés ainsi que l'architecture utilisée pour l'interface graphique.

2.2 Organisation du document

La suite de ce document donne une vue générale de l'architecture logicielle qui fait l'objet de ce travail.

Le paragraphe 2 rappelle les facteurs qualités attendus pour ce logiciel et explique comment mettre en œuvre des techniques pour atteindre ces facteurs. Le paragraphe 3 décrit le fonctionnement du framework XWiki et sa place

dans le développement de SECTra_w.

Le paragraphe 4 détaille les différentes vues architecturales (diagramme de contexte, vue logique, diagrammes de séquence).

Chapitre 3

Prise en compte des facteurs qualité attendus

3.1 Rappel des facteurs attendus

Le plan qualité logicielle définit trois exigences non fonctionnelles qui devront être validées :

- Utilisabilité : Aptitude du logiciel à pouvoir être utilisé facilement, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction.
- Maintenabilité : Aptitude du logiciel à pouvoir être corrigé facilement.
- Fiabilité : Aptitude d'un programme à accomplir sans défaillance l'ensemble des fonctions spécifiées dans un document de référence, dans un environnement opérationnel de référence, pour une durée d'utilisation donnée.

3.2 Application des tactiques

Les tactiques de conception architecturales permettent de définir un ensemble de règles de bonne conduite pour tendre vers les exigences non fonctionnelles requises du logiciel. Nous présenterons ces règles en fonction des facteurs de qualité. Le développement du logiciel devra s'appuyer sur ces règles. Nous présenterons ci-dessous 3 tactiques pour tendre vers un logiciel utilisable, maintenable et une pour la fiabilité en expliquant comment les appliquer à ce projet.

3.3 Utilisabilité

L'utilisabilité est autour de 3 questions :

- Le logiciel est-il facile à utiliser ?
- Le logiciel est-il facile à apprendre ?
- Le logiciel est-il facile à comprendre ?

La définition d'interface doit être claire et homogène, minimiser le nombre de clics nécessaires pour accéder à une fonction. Toutes les fonctions et les interfaces de logiciel doit absolument correspondre au manuel d'utilisateur.

3.4 Maintenabilité

3.4.1 Localiser les modifications

Les composants d'un même module doivent avoir une forte cohésion. Ceci permet une recherche efficace des modules à modifier et limite leur nombre. Le nombre de composants à modifier pour les évolutions les plus probables doit être minimal. Il faudra donc tenir compte des évolutions du logiciel.

3.4.2 Eviter les effets de bord

L'utilisation de données privées (accessibles uniquement par un composant) et la définition d'interface permet d'isoler les modifications au niveau des données privées tout en conservant l'interface. Les composants en interaction avec le composant modifié via une interface n'auront donc pas besoin d'être modifiés. Les composants doivent être le plus faiblement couplés possible entre eux, le nombre de communications devra donc être limité et celles ci seront des communications par valeur.

3.5 Fiabilité

La fiabilité concerne les problèmes de sécurité, qualité, durabilité et tolérance aux fautes. Donc, il faut bien définir les tests et minimiser la complexité de programme.

Chapitre 4

Le framework XWiki

SECTra_w (Système d'Exploitation de Corpus de Traductions sur le Web) est utilisé pour l'évaluation de systèmes de TA (traduction automatique), et comme "dorsal" dans des projets de post-édition collaborative de documents ou pages Web pré-traduits par un ou plusieurs systèmes de TA. Son extension iMAG (interactive Multilingual Access Gateway) permet de naviguer sur des sites Web dans diverses langues, grâce à l'appel de systèmes de TA gratuits, et à la possibilité d'améliorer les pré-traductions par "postédition collaborative", dans le contexte de lecture. À la différence de GoogleTranslate ou Systranet, tout "segment" (phrase ou titre) postédité apparaît immédiatement dans sa nouvelle version, qui est mémorisée dans une mémoire de traductions. Le tout est développé au-dessus de Xwiki, en utilisant html, groovy, velocity, et javascript.

Nous proposons d'utiliser un Wiki tel que XWiki, Tikiwiki, Twiki, Docuwiki, etc. comme plate-forme permettant de développer facilement et rapidement un système de support collaboratif au travail humain sur les corpus de traduction. Parmi ces Wikis, nous préférons XWiki.

- **Avantages.** La technologie Wiki a été largement utilisée pour développer des environnements collaboratifs, XWiki est un Wiki Open-Source écrit en Java, permettant la gestion des versions du code source et des contributeurs au niveau de la programmation. Il offre tout à la fois les fonctions principales d'un wiki (édition collaborative, suivi d'information, gestion de l'accès des membres) et des possibilités de développement avancées (avec l'aide de langages de programmation utilisables directement au sein du wiki). XWiki offre également la possibilité d'une gestion fine des droits d'utilisateurs, permettant de voir, d'éditer, de commenter, et d'administrer un

espace ou même une page. Cela permet de nombreuses configurations : public ou privé, librement éditable ou non, ou un mélange des deux.

- **Inconvénients.** Cependant, XWiki présente quelques inconvénients. Premièrement, XWiki ne permet la gestion de versions qu’au niveau des pages, alors que d’autres comme Docuwiki sont au niveau des paragraphes. Deuxièmement, XWiki n’est pas très compatible avec certains navigateurs tels que IE, car il utilise des fonctions évoluées (comme Ajax) qui ne sont pas encore standardisées sur les différents navigateurs.
- **Solutions.** Nous devons alors remodeler XWiki pour pouvoir l’adapter à un environnement de post-édition collaborative jusqu’au niveau des segments. Nous devons aussi trouver divers moyens permettant la communication et l’échange entre utilisateurs, et remodeler la gestion des utilisateurs pour l’adapter à nos besoins.

Chapitre 5

Architecture

5.1 Vue logique

5.1.1 Vue logique pour version 1

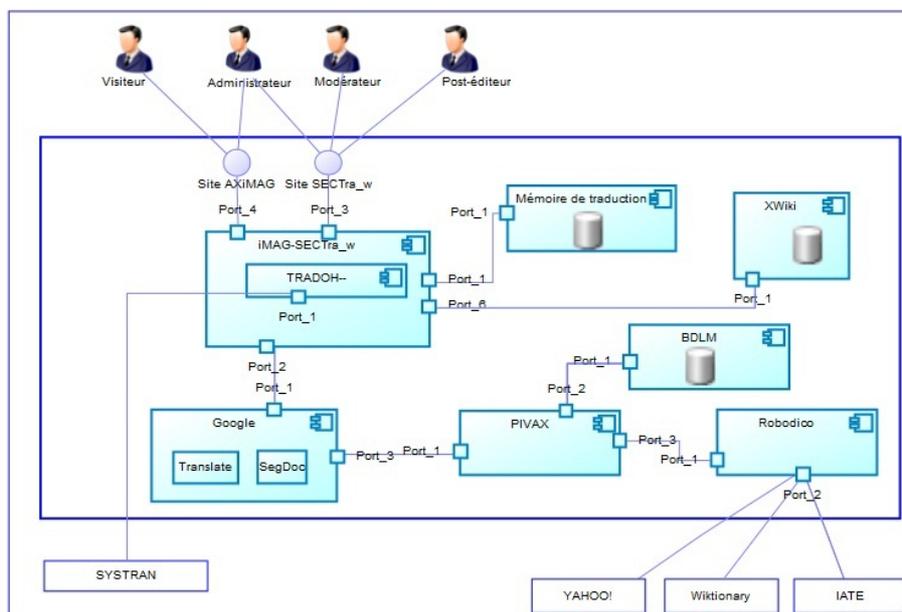


FIGURE 5.1 – Vue logique pour V1

5.1.2 Vue logique pour version 2

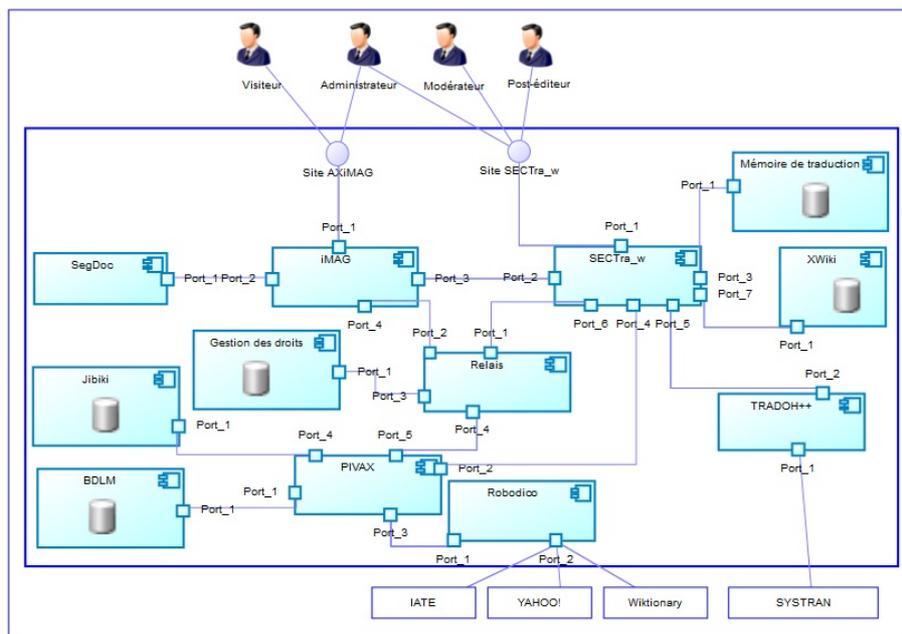


FIGURE 5.2 – Vue logique pour V2

5.1.3 Description des composants

Voir le document « Cahier Des Charges », paragraphe 5.2 « Composants ».

5.2 Vue physique

Version 1 :

Le logiciel est déployé sur une seule machine, un PC. L'IDE XWiki doit être installé. XWiki peut aussi s'exécuter sur tout conteneur de servlet récent (Tomcat, JBoss, GlassFish...) et avec plusieurs bases de données relationnelles (comme HSQL et MySQL). Pour ce tutoriel on utilisera Tomcat et MySQL. Voici versions utilisées dans ce tutoriel :

| Nom | Version |
|----------------------|---------|
| Sun-Java5-SDK | 1.5 |
| Tomcat 5 | 5.5.28 |
| MySQL 5 | 5.5.9 |
| Apache | 2.0.64 |
| PhpMyadmin | 3.3.9.2 |
| XWiki | 1.3 |
| mysql-connector-java | 5.1.5 |

Version 2 :



FIGURE 5.3 – Vue physique

5.3 Vue dynamique

Les diagrammes de séquence suivants représentent les cas d'utilisations décrits dans le dossier de spécifications externes et manuel d'utilisateur.

5.3.1 S'inscrire

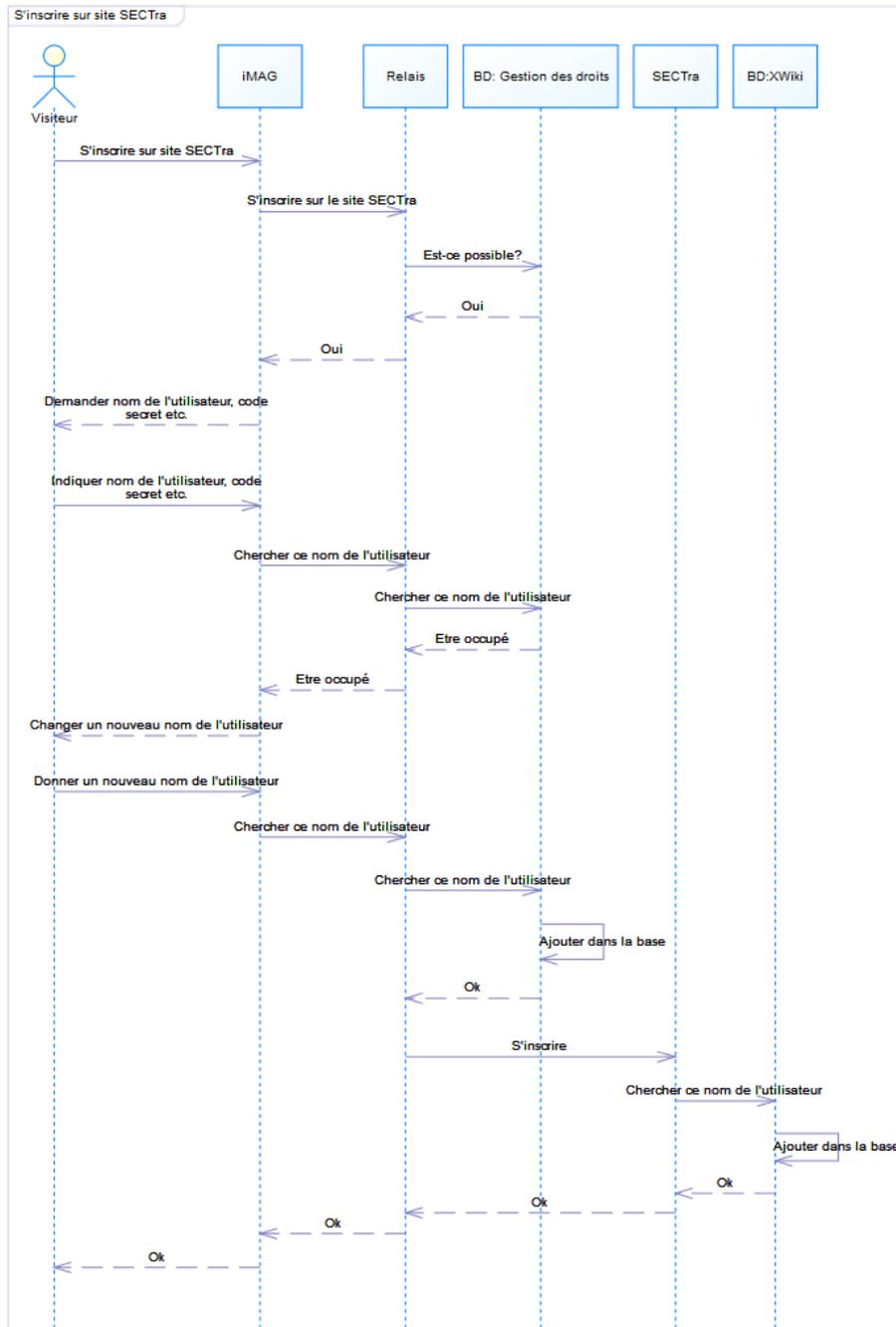


FIGURE 5.4 – S'inscrire sur site SECTra

5.3.2 Changer le code secret

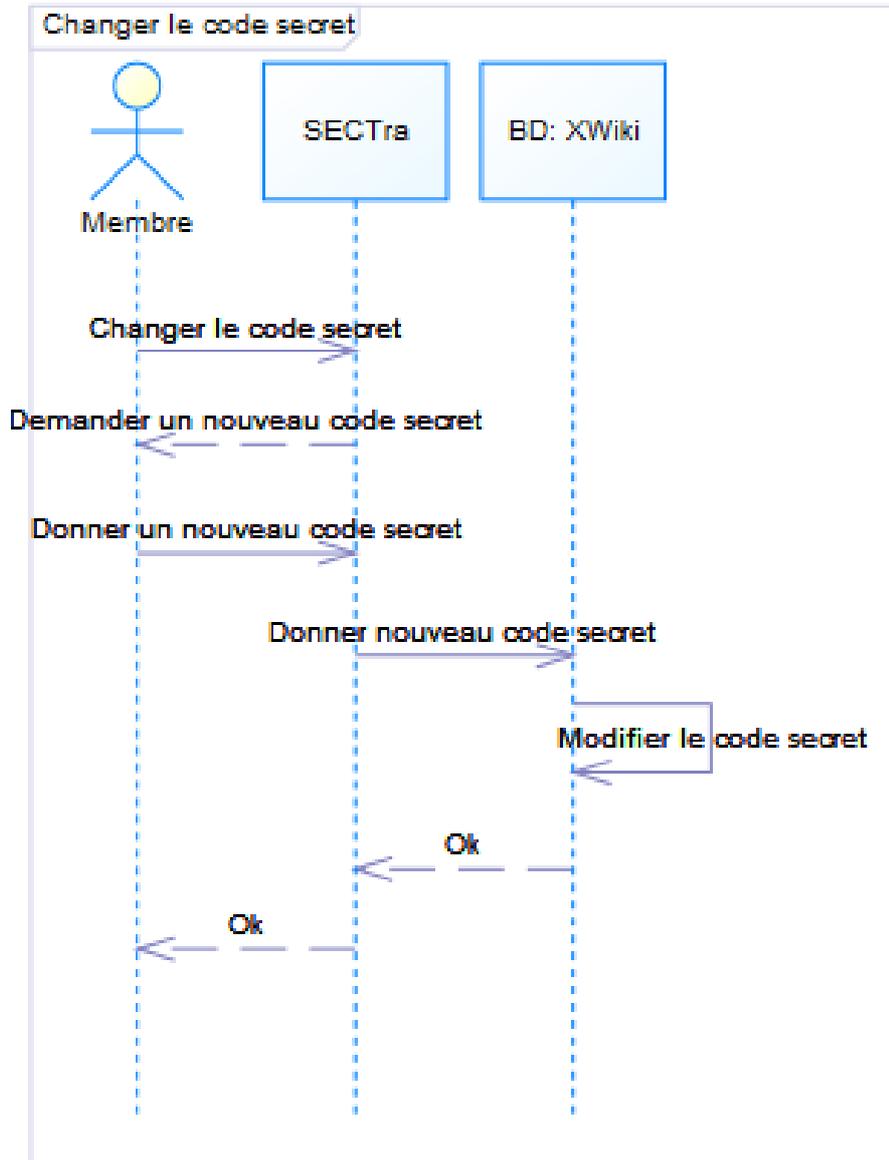


FIGURE 5.5 – Changer le code secret

5.3.3 Ajouter un corpus

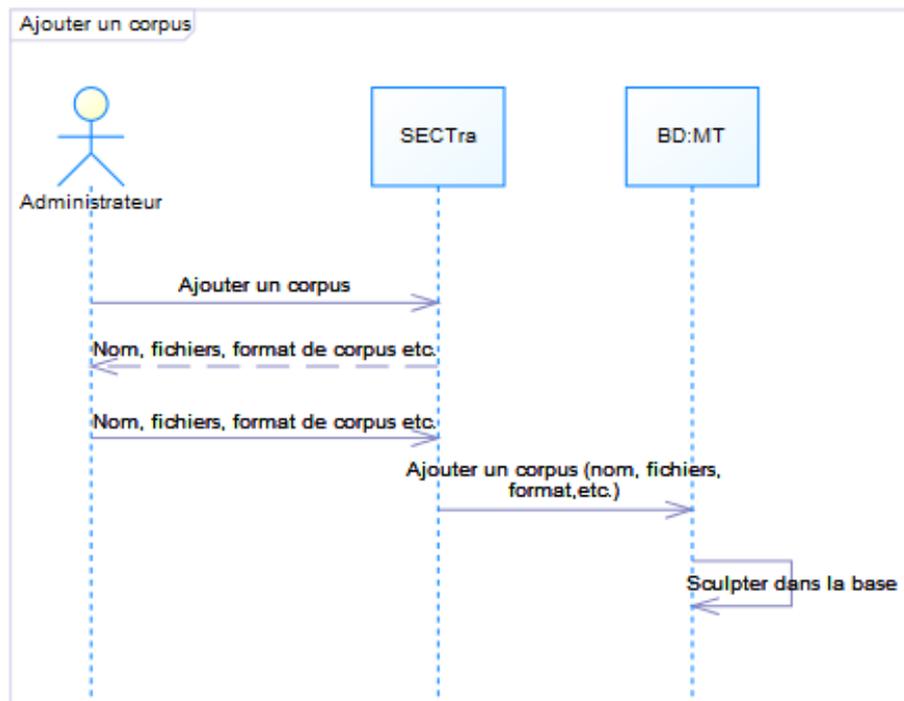


FIGURE 5.6 – Ajouter un corpus

5.3.4 Post-éditer

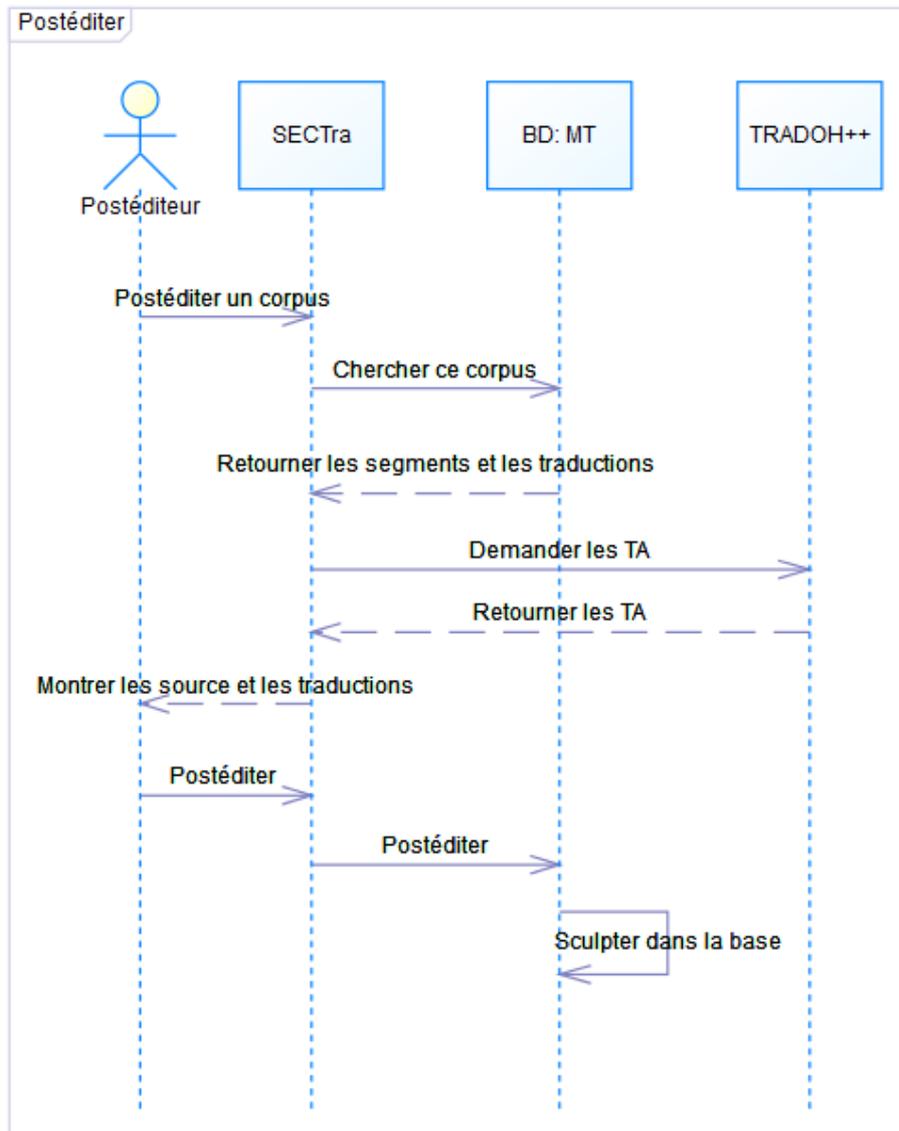


FIGURE 5.7 – Post-éditer

5.3.5 Utiliser dictionnaire

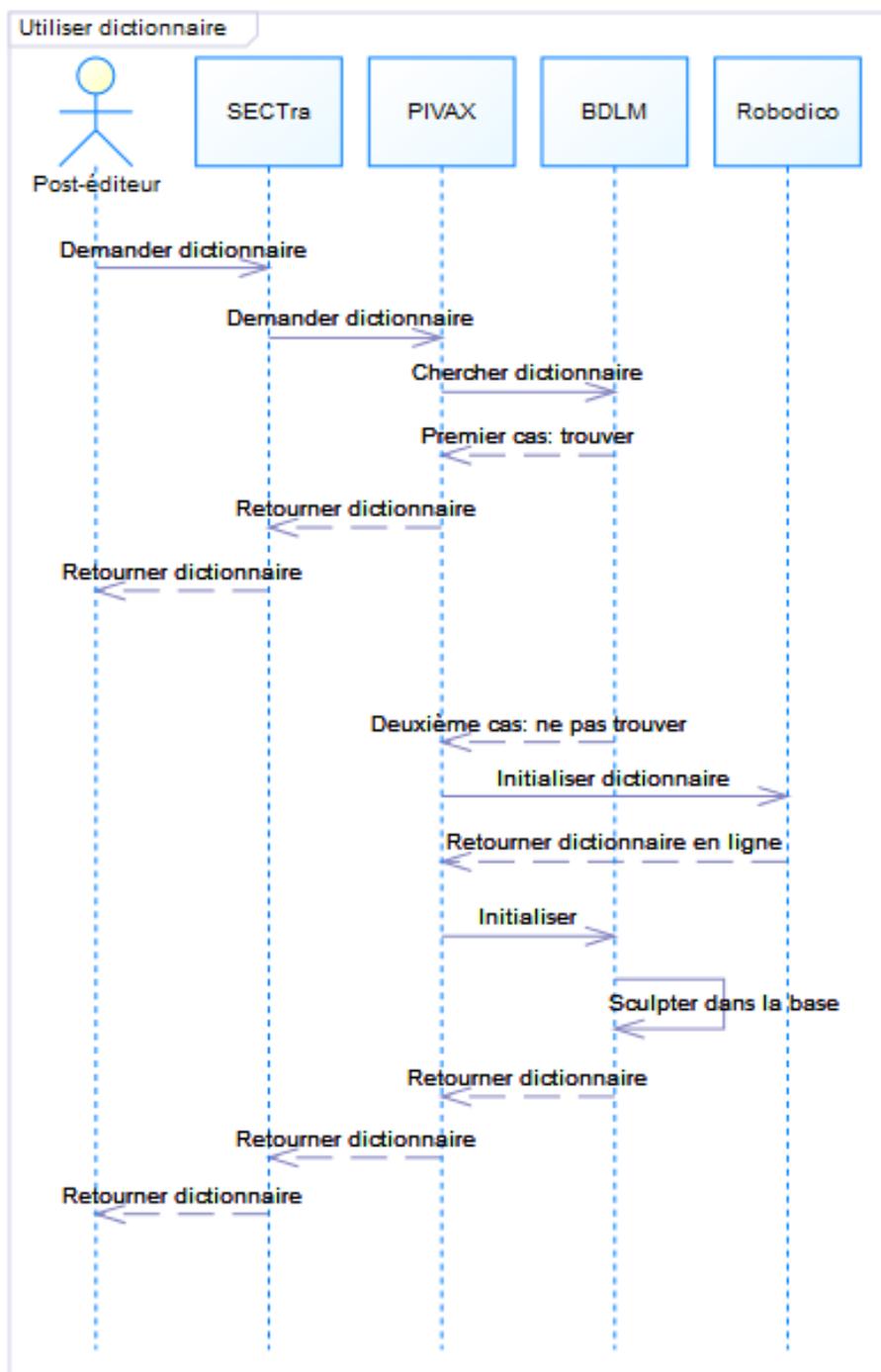


FIGURE 5.8 – Utiliser dictionnaire

5.3.6 Créer et initialiser un site en plusieurs langues

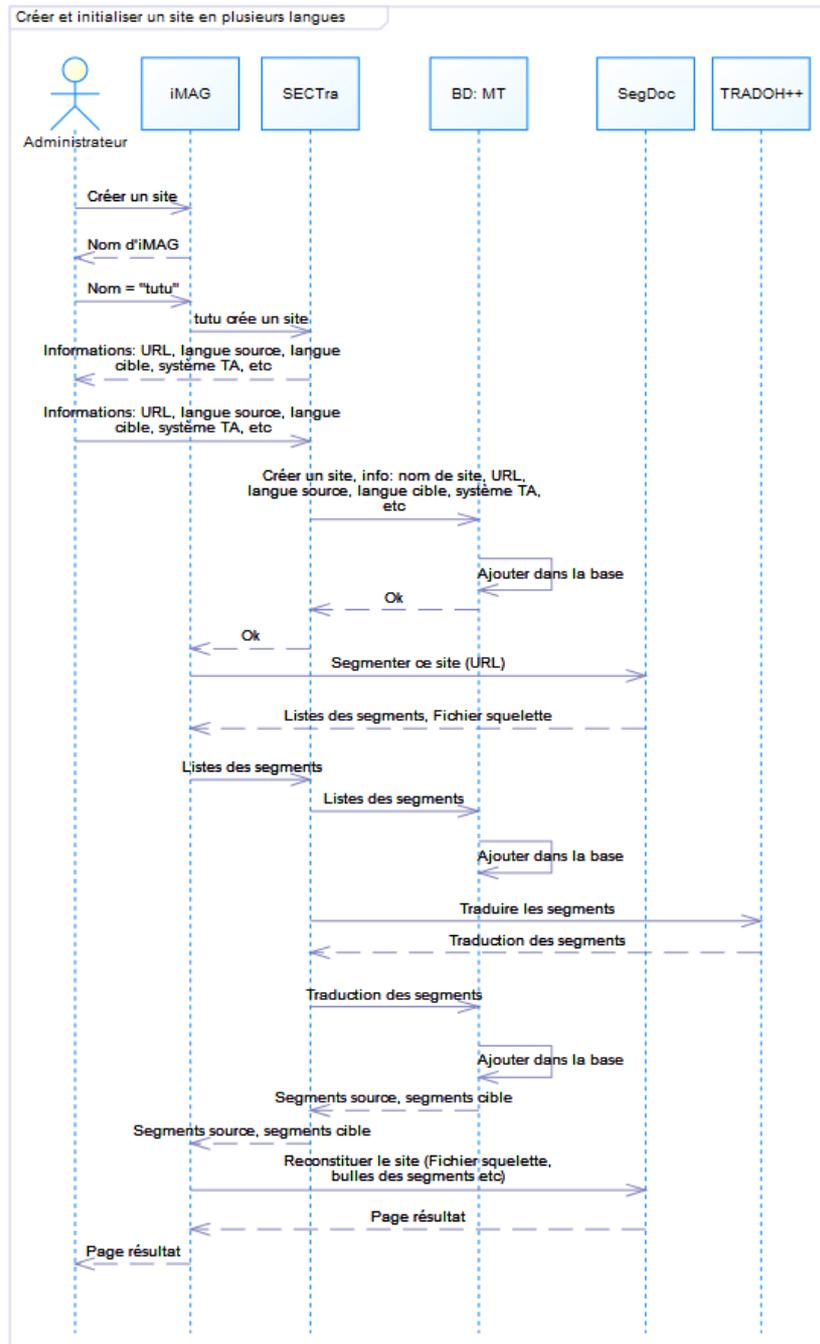


FIGURE 5.9 – Créer et initialiser un site en plusieurs langues

Index

- Ajouter un corpus, [14](#)
- Application des tactiques, [5](#)
- Architecture, [9](#)
- But du document, [3](#)
- Changer le code secret, [13](#)
- Créer et initialiser un site en plusieurs langues, [17](#)
- Description des composants, [10](#)
- Eviter les effets de bord, [6](#)
- Fiabilité, [6](#)
- Introduction, [3](#)
- Le framework XWiki, [7](#)
- Localiser les modifications, [6](#)
- Maintenabilité, [6](#)
- Organisation du document, [3](#)
- Post-éditer, [15](#)
- Prise en compte des facteurs qualité attendus, [5](#)
- Rappel des facteurs attendus, [5](#)
- S'inscrire, [12](#)
- Tableau de suivi des modifications, [2](#)
- Utilisabilité, [6](#)
- Utiliser dictionnaire, [16](#)
- Vue dynamique, [11](#)
- Vue logique, [9](#)
- Vue logique pour version 1, [9](#)
- Vue logique pour version 2, [10](#)
- Vue physique, [10](#)