

Table des matières

Table des matières	1
Table des figures	3
Liste des tableaux	5
Remerciements	1
Introduction générale	3
1 Cadre de projet	5
1 Introduction	5
2 Bibliothèque nationale de France et Gallica	5
3 Plan triennal de la recherche 2010-2013 à la BnF	6
4 Problématique du projet	6
5 Conclusion	7
2 Etat de l’art	9
1 Introduction	9
2 Numérisation des documents	9
3 Reconnaissance optique de caractère	10
3.1 Segmentation de la page du document	12
3.2 Reconnaissance de caractères	14
3.3 Le post-traitement	15
4 Les fichiers ALTO	16
5 Production des documents numériques à la BnF	18
5.1 Technique de lecture du fichier ALTO	21
6 Conclusion	22
3 Conception et Gestion de projet	23
1 Introduction	23
2 Gestion de projet	23
3 Expression des besoins et modele d’analyse	26
3.1 Langage et modélisation unifiée	26
3.2 Pourquoi on a choisi l’UML comme une méthode de concep- tion ?	26
3.3 Identification des acteurs	26
3.4 Diagramme de cas d’utulisation	27

3.5	Diagramme de séquence	27
4	Spécification des besoins	27
4.1	Diagramme de Cas d'utilisation général de l'application	28
4.2	Gestion des scénarios	31
4.3	Diagramme d'états-transitions	61
4.4	Diagramme de classe général de l'application	63
5	Conclusion	78
4	Réalisation	79
1	Introduction	79
2	Environnement de travail	79
2.1	Environnement matériel	79
2.2	Environnement de logiciel	79
3	Interfaces graphiques	81
3.1	Interface de contrôle Automatique	81
3.2	Interface de comparaison entre deux OCR	95
	Conclusion générale	99
	Bibliographie	101

Table des figures

2.1	La méthode X-Y Cut	14
2.2	Exemples de la classification Vox	15
2.3	Le traitement d'OCR	17
2.4	Encodage des principaux éléments de la page	19
2.5	Schéma de réalisation des marchés numérisation	21
3.1	Planning prévisionnel	25
3.2	Diagramme de Cas d'utilisation général de l'application	29
3.3	Diagramme de séquence (creation projet contrôle automatique côte image)	32
3.4	Diagramme de cas d'utilisation scénario1	32
3.5	Diagramme de Séquence scénario1	35
3.6	Diagramme de cas d'utilisation scénario2	36
3.7	Diagramme de séquence scénario2	38
3.8	Diagramme de cas d'utilisation scénario3	39
3.9	Diagramme de séquence scénario3	41
3.10	Diagramme de cas d'utilisation scénario4	42
3.11	Diagramme de séquence scénario4	44
3.12	Diagramme de cas d'utilisation scénario5	45
3.13	Diagramme de séquence scénario5	47
3.14	Diagramme de cas d'utilisation scénario6	48
3.15	Diagramme de séquence scénario6	50
3.16	Diagramme de sequence (creation de projet controle automatique cote repertoire d'images)	51
3.17	Diagramme de cas d'utilisation scénario7	52
3.18	Diagramme de séquence scénario7	54
3.19	Diagramme de cas d'utilisation scénario8	55
3.20	Diagramme de séquence scénario8	56
3.21	Diagramme de séquence (création projet Comparaison entre deux OCR côte repertoire d'images)	57
3.22	Diagramme de cas d'utilisation scénario9	58
3.23	Diagramme de séquence scénario9	60
3.24	Diagramme d'état-transitions de l'objet TextBlock	61
3.25	Diagramme de classe général de l'application	78
4.1	Le fonctionnement de Qt	80
4.2	Schéma de l'interface de contrôle automatique côte repertoire d'images	81

4.3	Interface de enregistrer sous le projet	82
4.4	Enregistrement du projet	82
4.5	Outils correction des strings	83
4.6	Avant l'ajout du mot dans l'application	83
4.7	Avant l'ajout du mot dans le fichier ALTO	83
4.8	Après l'ajout du mot dans l'application	84
4.9	Après l'ajout du mot dans le fichier ALTO	84
4.10	Avant la suppression du mot à travers notre l'interface de notre ap- plication	85
4.11	Avant la suppression du mot dans le fichier ALTO	85
4.12	Après la suppression du mot à travers l'interface graphique de notre application	85
4.13	Après la suppression du mot dans le fichier ALTO	85
4.14	Avant la modification du mot à travers l'interface graphique de notre application	86
4.15	Avant la modification du mot côté fichier XML	86
4.16	Après la modification du mot à travers l'interface graphique de notre application	87
4.17	Le résultat de l'opération de modification des mots qui existent dans le fichier ALTO	87
4.18	L'opération d'ajout d'un TextBlock à travers notre interface graphique	88
4.19	Le résultat de l'opération d'ajout des paragraphes dans le fichier ALTO	88
4.20	Opération d'ajout d'une ligne (TextLine) à travers l'interface gra- phique de notre application	89
4.21	Le résultat d'ajout des lignes dans le fichier ALTO	89
4.22	Ajouter string (phase de traçage de la boîte englobante d'un mot) . .	89
4.23	Ajouter string (phase d'annotation de la contenu d'un mot)	90
4.24	Résultats de l'opération d'ajout des mots dans le fichier ALTO. . . .	90
4.25	L'élément incorrect "TextBlock" dans le fichier ALTO	91
4.26	Suppression de la boîte englobante TextBlock	91
4.27	La position du TextBlock après suppression	91
4.28	l'affichage des boîtes englobantes Illustration avant l'opération de mo- dification	92
4.29	Le contenu du fichier ALTO avant l'opération de modification des coordonnées des boîte englobantes	92
4.30	Les boîtes englobantes "Illustrations" après l'opération de modifica- tion de ses coordonnées	92
4.31	Le contenu du fichier ALTO après l'opération de modification des boîtes englobant	93
4.32	Etape d'ouverture de l'outils	93
4.33	Etape de détermination du mot	93
4.34	Etape de détection du mot	94
4.35	Affichage TextBlock	94
4.36	Schéma de l'interface de comparaison entre deux OCR	95
4.37	Comparaison entre les TextBlocks de deux fichier ALTO (Etape 1) . .	96
4.38	Comparaison entre les TextBlocks de deux OCR (Etape 2)	96

Liste des tableaux

3.1	Tableau des Attributs de l'objet ALTO	64
3.2	Tableau des Attributs de l'objet Projet	64
3.3	Tableau des Attributs de l'objet Page	65
3.4	Tableau des Attributs de l'objet PrintSpace	65
3.5	Tableau des Attributs de l'objet TopMargin	65
3.6	Tableau des Attributs de l'objet composedBlock	66
3.7	Tableau des Attributs de l'objet TextBlock	66
3.8	Tableau des Attributs de l'objet TextLine	66
3.9	Tableau des Attributs de l'objet String	67
3.10	Tableau des Attributs de l'objet SP	67
3.11	Tableau des Attributs de l'objet Illustration	67
3.12	Tableau des Opérations de l'objet TextBlock	68
3.13	Tableau des Opérations de l'objet TextLine	70
3.14	Tableau des Opérations de l'objet String	70
3.15	Tableau des Opérations de l'objet SP	71
3.16	Tableau des Opérations de l'objet Illustration	72
3.17	Tableau des Opérations de l'objet ALTO	72
3.18	Tableau des Opérations de l'objet Projet	73
3.19	Tableau des Opérations de Contrôleur String	73
3.20	Tableau des Opérations de Contrôleur Segmentation	74
3.21	Tableau des Opérations de Contrôleur Comparaison	75
3.22	Tableau des Opérations de Contrôleur projet	75
3.23	Tableau des Opérations de l'interface graphique	76
3.24	Tableau des Opérations de l'interface Comparaison entre deux OCR	77

Remerciements

C'est avec plaisir que je réserve ces quelques ligne en signe de gratitude et de profonde reconnaissance à tous ceux qui, de près ou de loin, ont contribué a l'aboutissement de ce travail.

Au terme de ce travail , je voudrais bien exprimer ma gratitude à mon encadreur **Monsieur Mohamed Tmar** pour avoir accepté, d'aussi bonne grâce, d'assurer le suivi de ce projet de fin d'études.

je saisis aussi cette occasion pour remercier les membres du jury d'avoir accepté de juger ce travail et j'espère qu'ils trouveront dans ce rapport les qualités de clarté et de motivation qu'ils attendent.

j'adresse aussi mes remerciements à Monsieur **Ahmed Ben Salah** pour m'avoir encadré et pour m'avoir donné l'occasion de travailler sur ce sujet riche,actuel et passionnant. Sa grande disponibilité son sens des relations , sa regueur , son experience , sa pidagogie et ses critiques constructives m'ont été prcieux.

Je Tenaiss aussi à remercier **Monsieur Mohamed Ali Chakroun** pour ses conseils frutueux pour bien compléter le projet.

Ma gratitude pour tous les membres de la bibliotheque Nationale de France et surtout **Mme Genevieve Cron** pour la bonne tout au long de ce projet.

Introduction générale

Depuis 1991, la Bibliothèque nationale de France (BNF) a introduit plusieurs projets de numérisation afin de conserver les documents et les diffuser. Le processus de numérisation commence par une étape de programmation annuelle des documents à numériser puis par une phase de sélection des documents à reconnaître et enfin par l'envoi de ces documents au prestataire de numérisation.

La numérisation de documents est une étape importante dans la mise en place d'un système de gestion électronique de documents (GED). Le choix de la solution de numérisation doit prendre en compte la qualité du document original ainsi que toutes les étapes de traitement des documents depuis l'acquisition, la conversion du contenu jusqu'à la correction et la mise en exploitation du document final. Le but de cette numérisation est une utilisation du contenu converti, par exemple pour effectuer une recherche d'information.

Plusieurs facteurs agissent sur la qualité finale des résultats de conversion de l'OCR, la présence d'un outil automatique pour détecter les défauts de l'image du document, peut être un bon moyen pour nous guider à choisir les techniques de l'OCR et pour prédire le taux de conversion.

C'est dans le cadre des application de traitement des documents numerique que notre projet de fin d'étude s'intègre.

La bibliotheque nationale de France nous a confié de contribuer à la réalisation d'une application qui a pour objectif d'offrir une solution informatique, basée sur de nouvelles méthodes ergonomiques de controle d'OCR et nommée **BnF OCR Control**. Plus particulièrement, il s'agit d'une application pour les contrôleurs de la BNF permettent de faciliter la tâche de la correction et de contrôle des documents numériques.

Dans le premier chapitre, intitulé **Cadre de projet** nous présentons l'organisme d'accueil, le sujet proposé et les principales solutions existantes. Le deuxième chapitre est dédié à **l'Etat de l'Art** dans lequel nous définissons certaines technologies utilisées dans notre application. La troisieme est consacré à la **Conception** de la solution et enfin le dernier chapitre décrit l'architecture, le modèle d'implémentation ainsi que les résultats de test et de validation.

Chapitre 1

Cadre de projet

1 Introduction

Il est difficile de parler d'un projet avant d'avoir fait une analyse détaillée du travail à faire. Il est cependant nécessaire de faire une première estimation générale pour pouvoir cadrer le projet, le lancer et le vendre.

2 Bibliothèque nationale de France et Gallica

La bibliothèque nationale de France (BnF) ainsi dénommée depuis 1994, est la bibliothèque nationale de la République française, héritière des collections royales constituées depuis la fin du Moyen Âge. Elle est la plus importante bibliothèque de France et l'une des plus importantes au monde. Elle a le statut d'établissement public. La bibliothèque nationale de France comporte quatorze départements et plusieurs collections principalement conservées sur ses quatre sites parisiens, dont le Cabinet des Médailles. Hors de Paris, elle comprend la maison Jean-Vilar à Avignon et deux centres techniques de conservation à Bussy-Saint-Georges et Sablé-sur-Sarthe.

La collection documentaire de la BnF est très riche culturellement. Elle regroupe des documents avec différentes langues et qui appartiennent à différents dictionnaires de langue. Les caractéristiques physiques de cette collection sont très hétérogènes. En effet, puisque les livres de la BnF appartiennent à plusieurs âges d'imprimerie nous trouvons plusieurs genres de papier et différents types d'encre qui sont présents dans cette collection. Malheureusement, pour des raisons de sécurité l'accès grand public à ces documents précieux et anciens est limité. En effet, leur état physique délicate augmente les risques d'endommagement de ces documents.

Pour des raisons de conservation, la bibliothèque nationale de France a lancé depuis 1992 des projets de numérisation de masse des documents et pour rendre sa collection accessible au lecteur la BnF a lancé en 1997 sa bibliothèque numérique nommée Gallica. Aujourd'hui, elle est la plus grande bibliothèque numérique à l'échelle mondiale. Cette bibliothèque est en libre accès, elle regroupe des livres numérisés, des cartulaires, des revues, des photos et une collection d'enluminures. Le 10 février 2010, Gallica a franchi le cap du millionième document avec Scènes de

la vie de Bohème, d'Henry Murger, 1913. Au 24 février 2010, Gallica proposait à la consultation en ligne 1 020 766 documents dont 408 190 en mode texte : 184 157 livres, 5 462 périodiques, revues et journaux (soit 698 446 fascicules), 120 102 images fixes, 4 722 manuscrits, 9 759 cartes et plans, 2 523 partitions et 1 057 documents sonores, soit un rythme de 1 500 documents numérisés par jour. Un certain nombre d'ouvrages a fait l'objet d'une Reconnaissance optique de caractères et le texte peut être recherché sur Gallica.

3 Plan triennal de la recherche 2010-2013 à la BnF

Le plan triennal de recherche est un plan de recherche lancé depuis 1994 par la Bibliothèque nationale de France afin de conduire des programmes de recherche autour des domaines qui concernent les documents. Cette activité répond à des exigences scientifiques précises en termes d'obligation de résultats, de travail en partenariat et de programmation rigoureuse dans le temps. Les domaines concernés par ce plan de recherche sont :

- **Bibliographie**
- **Écrit**
- **Iconographie**
- **Livre**
- **Musique**
- **Numismatique**
- **Technologies**

Dans le domaine des technologies d'OCR la BnF a lancé un projet de recherche en partenariat avec l'Université de Rouen et l'université François Rabelais de Tours afin de maîtriser les technologies de l'OCR et de contrôle des textes numériques. Notre travail entre dans le cadre d'une thèse qui appartient à ce plan de recherche et dans lequel nous avons réalisé une interface graphique qui permet d'incorporer les algorithmes de contrôle automatique des résultats de l'OCR. L'interface produite offre aussi les moyens de correction manuelle des résultats de l'OCR et de comparaison entre deux résultats d'OCR.

4 Problématique du projet

La productions des documents numériques à la BnF se fait à l'aide des prestataires de numérisation des documents. Dans les projets de numérisation de masse le contrôle de qualité des documents numériques nécessite des outils adéquats et

ergonomiques pour faciliter la tâche de correction et de contrôle. Ce genre d'outil n'est pas encore présent à la BnF ce qui complique la tâche de contrôle de qualité.

D'autre part, le processus de contrôle de l'OCR à la BnF se base sur le contrôle d'une échantillon de pages choisi de façon aléatoire selon la norme ISO. Cette méthode limite l'opération de contrôle de l'OCR sur un nombre de pages de document et pas sur la totalité des pages du document.

L'interface que nous avons réalisée dans le cadre de notre projet de fin d'étude répond bien aux besoins de la BnF. En plus, elle est capable d'inclure les algorithmes de contrôle automatique produits dans le cadre du projet de recherche.

5 Conclusion

Dans ce chapitre on a défini le problématique du projet et certain objectifs à atteindre au cours de réalisation du projet. Dans le deuxième chapitre on va implémenté quelque définition sur le domaine de travail.

Chapitre 2

Etat de l'art

1 Introduction

Faire l'état de l'art consiste à rechercher toutes les informations, publications formelles ou informelles, découvertes, nouveautés, et inventions sur toutes les dernières avancées scientifiques, techniques, économiques ainsi que sur les travaux antérieurs ayant un lien avec le domaine sur lequel on s'apprête à travailler.

2 Numérisation des documents

La numérisation des documents est une étape importante dans la mise en place d'un système de gestion électronique de documents (GED) puisqu'elle permet de passer de l'état physique à l'état numérique du document. L'objectif de la numérisation des documents est l'utilisation du contenu converti, par exemple pour effectuer une recherche d'information. La chaîne de numérisation contient plusieurs étapes qui sont :

- **L'acquisition** : permettant la conversion du document papier sous la forme d'une image numérique (bitmap). Cette étape est importante car elle se préoccupe de la préparation des documents à saisir, du choix et du paramétrage du matériel de saisie (scanner), ainsi que du format de stockage des images.

- **Le prétraitement** : dont le rôle est de préparer l'image du document au traitement. Les opérations de prétraitement sont relatives au redressement de l'image, à la suppression du bruit et de l'information redondante, et enfin à la sélection des zones de traitement utiles.

- **La reconnaissance automatique du contenu** : qui conduit le plus souvent à la reconnaissance du texte et à l'extraction de la structure logique. Ces traitements s'accompagnent le plus souvent d'opérations préparatoires de segmentation en blocs et de classification des médias (graphiques, tableaux, images, etc.). Les mots formés après la phase de reconnaissance sont envoyés à un dictionnaire de mots pour vérifier son exactitude.

• **La correction manuelle des résultats :** est opérée quand un taux de reconnaissance élevé est demandé. Cette opération se réalise par une comparaison des résultats de l'OCR avec la vérité terrain à travers des interfaces dédiées.

L'acquisition du document est opérée par un système de balayage optique appelé scanner. Le résultat est rangé dans un fichier de points appelé image. Les pixels peuvent avoir la valeur : 0 ou 1 pour les images binaires ou une plage de valeurs entre 0 et 255 pour les images en niveau de gris. La résolution est exprimée en nombre de points par pouce. Les valeurs courantes utilisées couramment vont de 100 à 400ppp. Par exemple, en 200ppp, la taille d'un pixel est 0.12mm ce qui représente 8 points par mm. Pour un format classique A4 et une résolution de 300ppp, le fichier image contient 2520×3564 pixels. La technicité des matériels d'acquisition (scanner) a fait un progrès considérable ces dernières années. On trouve aujourd'hui des scanners pour des documents de différents types (feuilles, revues, livres, photos, etc.). Leur champ d'application va de la digitalisation de textes à la digitalisation de photos en 16 millions de couleurs (et même plus pour certains). La résolution est classiquement de l'ordre de 300 à 960ppp selon les modèles.

3 Reconnaissance optique de caractère

La reconnaissance optique des caractères est réalisée à l'aide de systèmes dédiés appelés OCR. L'OCR (Optical Character recognition) est un système qui analyse les images des documents imprimés ou manuscrits pour donner un fichier texte. La tâche d'un OCR consiste à segmenter les images en lignes, mots, caractères puis effectuer la reconnaissance des symboles. Actuellement plusieurs systèmes de reconnaissance optique des caractères sont utilisés dans plusieurs domaines :

- Les banques pour l'authentification des chèques et les assurances pour la vérification de clauses de contrats.
- La poste pour la lecture des adresses et le tri automatique du courrier.
- Les bibliothèques pour faciliter l'accès aux documents anciens et précieux.

Par contre, il n'existe pas un système d'OCR complet qui marche sur tous les types de document mais il existe plutôt des systèmes qui dépendent du type des données traitées et de l'application envisagée. Le choix de technique d'OCR que nous devons utiliser se fait sur un certain nombre de critères physiques de document.

La qualité de l'image joue un rôle important dans la qualité du rendu final de l'OCR. En effet, le résultat de l'analyse des documents va être dépendant de la qualité de l'image de document acquise.

Les systèmes de l'OCR commencent leurs analyses par un prétraitement tabulaire qui divise l'image des documents en plusieurs parties (textuelles, graphiques, etc.), puis en sous parties (lignes, caractère, etc.). Après la division de l'image en plusieurs blocs, le moteur de l'OCR essaye de reconnaître les formes des caractères détectés en comparant chaque forme avec les formes qui existent déjà dans la base de données des caractères.

Un texte est une association de caractères appartenant à un alphabet, réunis

dans des mots d'un vocabulaire donné. L'OCR doit retrouver ces caractères, les reconnaître d'abord individuellement, puis les valider par reconnaissance lexicale des mots qui les contiennent. Cette tâche n'est pas triviale car si l'OCR doit apprendre à distinguer la forme de chaque caractère dans un vocabulaire de taille souvent importante, il doit en plus être capable de la distinguer dans chacun des styles typographiques (polices), chaque corps et chaque langue, proposés dans le même document. Cette généralisation omni-fonte et multilingue n'est pas toujours facile à cerner par les OCRs et reste génératrice de leurs principales erreurs.

Après l'étape de reconnaissance des caractères, le moteur de l'OCR vérifie l'exactitude des mots formés en utilisant un dictionnaire des mots adapté à la langue du document scanné. La combinaison des systèmes de reconnaissance automatique des caractères et de vérification des mots donne des résultats intéressants en termes de précision.

Pour conclure, un système de reconnaissance de textes est composé de plusieurs modules :

- **La segmentation** permet d'isoler les éléments textuels, les mots et les caractères, pour la reconnaissance. Elle se base sur des mesures de plages blanches (interlignes et inter caractères) pour faire la séparation. La multiplicité des polices et la variation des justifications empêchent de stabiliser les seuils de séparation conduisant à la génération des blancs inexistants ou au contraire à l'ignorance des blancs séparateurs des mots. Ce type d'erreur est très fréquent, d'après [1].

- **La reconnaissance de caractères** permet de se prononcer sur l'identité d'un caractère à partir d'un apprentissage de sa forme. Cette étape nécessite une étape préalable de paramétrisation de la forme, définissant des données, des mesures, ou des indices visuels sur lesquels s'appuie la méthode de reconnaissance. Suivant la nature de ces informations, il existe plusieurs catégories de méthodes : syntaxique (description par une grammaire), structurelle (description par un graphe), ou statistique (description par partitionnement de l'espace). Ces dernières ont de loin le plus grand intérêt avec les méthodes à base de réseaux de neurones, ou de modèles stochastiques. La complexité de la tâche vient de l'apprentissage qui nécessite, pour sa stabilité, un grand nombre d'échantillons par classe, et une recherche d'indices visuels discriminants, ce qui n'est pas aisé dans un contexte omni-fonte comme celui concerné par la numérisation automatique.

- **Le post-traitement** est effectué lorsque le processus de reconnaissance aboutit à la génération d'une liste de lettres ou de mots possibles, éventuellement classés par ordre décroissant de vraisemblance. Le but principal est d'améliorer le taux de reconnaissance en faisant des corrections orthographiques ou morphologiques à l'aide de dictionnaires de digrammes, trigrammes ou n-grammes. Quand il s'agit de la reconnaissance de phrases entières, on fait intervenir des contraintes de niveaux successifs : lexical, syntaxique ou sémantique.

3.1 Segmentation de la page du document

La segmentation de la structure physique consiste à localiser toutes les zones contenant des données homogènes. Ces zones sont souvent espacées et forment des blocs géométrique élémentaires, à base de rectangles dans la grande majorité des cas. La segmentation a pour fonction de distinguer les diverses composantes d'un document. Cela recouvre plusieurs problèmes :

- délimiter les caractères, les mots et les lignes de texte (trouver leur enveloppe rectangulaire)
- distinguer et délimiter les grandes zones d'information dans une page : texte, dessins et graphismes au trait, formules mathématiques, tableaux, photos

Selon [2], Il n'existe pas de méthode de segmentation de la structure physique qui soit générique à toutes les classes de documents. Cependant, on peut développer des méthodes de segmentation de la structure physique pour chaque catégorie de documents qui présentent une certaine homogénéité. On distingue plusieurs catégories de documents, du point de vue de l'analyse d'image, permettant d'utiliser des familles d'algorithmes de segmentation suivant les critères suivants :

- La nature des textes (imprimés, manuscrits ou mixtes).
- La complexité de la structure physique de la page qui concerne la feuille de style multicolonne du document et les documents composites incluant à la fois du texte et des images.
- La qualité de la résolution des images qui contient les informations nécessaires qui permettra une meilleure séparation entre les différentes composantes de l'image.

Les méthodes d'analyse et segmentation des images pour l'obtention d'une structure physique diffèrent suivant la stratégie de reconnaissance ascendante ou descendante choisie. La structure physique peut être obtenue soit par une approche ascendante guidée par des données, soit par une approche descendante guidée par un modèle. Les méthodes interactives reposent à la fois sur des méthodes descendantes de recherche d'information et sur des méthodes ascendantes de segmentation guidée par le contenu de l'image.

Segmentation ascendante

La catégorie des algorithmes de segmentation ascendante est caractérisée par le fait que l'analyse part de composants de bas niveau (comme les pixels, objets connexes, groupe de pixels faiblement espacés) récursivement pour essayer de les fusionner en utilisant des heuristiques de fusion de blocs de proche en proche. Elle nécessite donc des temps de calcul considérables du fait de la manipulation des milliers d'objets à partir d'une image à pleine résolution. Par contre ces méthodes sont simples à mettre en œuvre et ne nécessitent pas un modèle spécifique au type du document traité. Seules quelques règles élémentaires de fusion sont nécessaires pour extraire la structure physique de la page.

Les méthodes de segmentation ascendante sont applicables sur une grande va-

riété de documents mais ses résultats sont généralement imprécis ce qui nécessite des règles d'affinage.

Selon [3], il existe de nombreuses techniques de segmentation utilisant l'approche ascendante. Toutes n'ont pas les mêmes performances. La méthode RLSA introduite par Wahl, Wong et Casey dans [4] (Run Length Smoothing Algorithm) est très utilisée car relativement simple à mettre en œuvre.

Cette méthode est basée sur des opérations morphologiques de traitement d'image. Le principe de cette méthode est de noircir toute séquence de pixels blancs comprise entre deux pixels noirs, de longueur inférieure à un seuil donné.

En pratique l'algorithme est appliqué horizontalement et verticalement sur l'image binaire originale en répétant la procédure avec des seuils de lissage horizontal et vertical différents, extraire itérativement les blocs de l'image, puis les lignes de texte et les mots. Ces seuils de lissage sont les seuls paramètres de l'algorithme RLSA.

☞ Segmentation descendante

La famille de techniques de segmentation descendante essaie d'avoir une approche globale pour affiner les régions. Ces méthodes reposent sur un découpage récursif de l'image en analysant plutôt les espaces que les traits. D'après [2], ces méthodes sont extrêmement rapides car ils ne traitent pas les objets élémentaires de l'image. Elles sont adaptées aux documents avec une feuille de style régulière et bien reconnue avant l'OCR-isation des documents.

La segmentation descendante contient plusieurs méthodes. La plus connue d'entre elles est la méthode X-Y Cut, présentée par Nagy et Seth dans [NAG 86a]. Elle consiste à découper récursivement l'image du document à l'aide de l'analyse des fréquences des pixels noirs représentant l'encre. Cela signifie que cette technique utilise la projection horizontale et verticale de l'image du document afin de trouver les espaces interligne et découper les parties de l'image. Ces projections atteignent des pics le long des lignes de texte et forment des vallées autour des espaces de séparation entre les blocs textes (**voir Figure 2.1**). La séparation entre les zones de textes et les espaces inter blocs textuelles se fait à l'aide d'un seuil avec lequel toutes les zones qui contiennent des projections supérieures à ce seuil sont considérées comme du texte et toutes les zones qui contiennent des projections inférieures à ce seuil sont considérées comme des zones inter blocs textuels.

Le découpage de l'image de document se fait selon plusieurs niveaux : paragraphes, lignes, mots et caractères. A chaque niveau l'algorithme utilise un seuil qui correspond à la taille de l'espace blanche qui sépare les blocs textes. Le choix de ces seuils représente la grande limite de cette méthode.

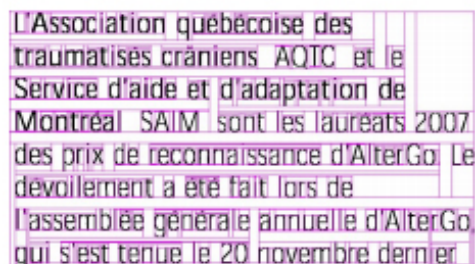


FIGURE 2.1 – La méthode X-Y Cut

☞ Les Approches mixtes

Les méthodes de segmentation ascendantes et descendantes ont chacun des avantages et des inconvénients. Les méthodes de segmentation ascendantes sont plus adaptées aux documents avec une structure physique variable, alors que les méthodes de segmentation descendantes sont utilisables que sur des documents avec une structure physique bien ordonnée ou pour lesquels la feuille de style est connue. Les approches mixtes essaient de tirer les avantages des deux méthodes. Les approches mixtes permettent de ne pas partir de tous les objets élémentaires de l'image ce qui les rend plus rapides que les approches ascendantes. De plus, les approches mixtes permettent de traiter des documents moins contraintes que pour les méthodes descendantes.

Selon [11], **Split and merge** est la méthode la plus connue dans cette catégorie d'approche. Elle donne de bons résultats de segmentation sur les pages bien structurées. Elle se déroule en deux étapes : une étape de découpage suivie d'une méthode d'agrégation. L'étape de découpage consiste à séparer en quatre régions l'image. elle s'arrête lorsque le bloc considéré est suffisamment homogène, le but étant de découper l'image en blocs de plus en plus petits, en fonction de l'homogénéité de l'image. Cette étape produit alors une image sur-segmentée. L'étape d'agrégation permet alors de fusionner les régions homogènes qui auraient été séparées lors de l'étape précédente.

3.2 Reconnaissance de caractères

L'opération de reconnaissance consiste à affecter à l'image d'un caractère inconnu fournie en entrée du système la classe qui lui correspond parmi un ensemble de classes connues. Après la phase de repérage des zones de texte, il faut identifier les caractères, les coder et les regrouper en mots : c'est en cela que consiste la reconnaissance de caractères. Cette opération fait appel à des techniques diverses ; la complexité du problème provient de la grande diversité des formes des caractères et des tailles des caractères imprimés, ainsi que des défauts d'impression et de numérisation. Un système de reconnaissance de caractères accepte les données de sortie provenant d'un équipement en ligne ou hors ligne comme des données d'entrée, en assure le traitement et produit des données de sortie compréhensibles. Un système de reconnaissance de caractères peut être effectivement morcelé en plusieurs composants. L'un de ces composants se charge des fonctions de « **pré-traitement** »,

comme la normalisation et l'amincissement. Une fois que la forme d'entrée a été pré-traitée, un autre composant l'accepte et en extrait les attributs caractéristiques. Les caractéristiques ainsi extraites sont utilisées par un composant de «**classification**» (voir Figure 2.2).

La phase de l'extraction des caractéristiques est très importante dans les systèmes de reconnaissance de caractères puisqu'elle vise à obtenir un ensemble pertinent de caractéristiques qui permettent de distinguer les formes de caractères. On peut distinguer les caractéristiques continues définies à partir des moments, de développement en séries, les caractéristiques construites, à partir de l'analyse des contours et des projections, enfin, les caractéristiques structurelles extraites à partir du squelette, telles que les fins de traits, les jonctions, les occultations....

L'étape de classification doit déterminer la classe inconnue du caractère à partir de son vecteur de caractéristiques. Selon le mode de fonctionnement de notre système et de la nature des caractéristiques extraits les concepteurs de l'OCR choisissent la méthode de classification des caractéristiques. Parmi les approches de classification les plus utilisées actuellement, on trouve les classificateurs k plus proches voisins, les classificateurs bayesiens, les réseaux de neurones. A la fin de cette opération nous obtenons un ou plusieurs classes par image de caractère.

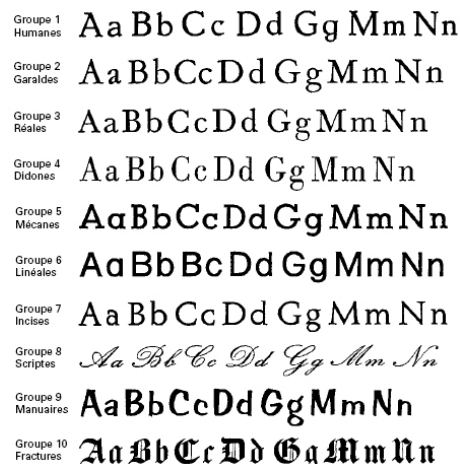


FIGURE 2.2 – Exemples de la classification Vox

3.3 Le post-traitement

Le post-traitement comprend la vérification, l'exécution de l'action et l'adaptation. L'objectif de la vérification est d'accroître le niveau de confiance dans la classification effectuée, une telle vérification peut être effectuée de diverses façons. L'une de ces façons consiste à utiliser une base de données comportant des combinaisons de 2 ou 3 lettres pour vérifier si la séquence des lettres reconnues ne comprend pas de combinaisons impossibles. Une autre possibilité est d'utiliser un dictionnaire pour vérifier si une certaine séquence de caractères constitue un mot valide.

En règle générale, cette méthode est moins fiable puisque les mots exacts qui ne sont pas consignés au dictionnaire sont rejetés. En plus des dictionnaires contenant des lettres et/ou des mots, un modèle grammatical formel de niveau plus élevé peut être utilisé pour vérifier l'exactitude d'expressions ou de phrases entières.

4 Les fichiers ALTO

Selon [?], dans le cadre du projet européen META-E, la BNF a coopéré avec la Bibliothèque du Congrès Américain pour développer le format ALTO.

Le format ALTO (Analyzed Layout and Text Object)¹ est un schéma XML qui contient des métadonnées qui décrivent le contenu et la structure physique d'une ressource textuelle comme les documents et les journaux.

Après les traitements de l'OCR, le fichier ALTO est utilisé pour stocker les informations sur la disposition et le contenu de n'importe quel document imprimé. Le cas le plus fréquent étant de pouvoir superposer l'image et son contenu converti en texte afin de mettre en surbrillance les mots trouvés à l'issue d'une requête. Il permet aussi de récupérer le flux textuel seul et de faire une mise en page via une feuille de style XSL comme pour tout fichier xml. Enfin le contenu textuel peut être exploité en base de données, ou pour accéder au contenu du document via une requête sur Internet.

Comme nous avons vu dans la section ??, le traitement d'OCR passe par une phase de découpage de l'information appelée segmentation. En effet pour reconnaître correctement les caractères qui constituent un mot, il faut repérer la ligne dans laquelle il se trouve dans un paragraphe présent dans une page imprimée. L'outil de segmentation dessine sur l'image à convertir des boîtes qui encadrent les divers blocs d'information avec d'une part les éléments textuels et d'autre part les éléments non textuels tels que les illustrations et les graphiques (**voir Figure 2.3**). Chaque bloc textuel est ensuite découpé en lignes afin que l'OCR puisse travailler sur les caractères pour reconstituer des mots. La dimension de chaque bloc de segmentation est calculée à partir des pixels correspondants sur l'image.

Les informations suivantes sont fournies dans chaque fichier ALTO produit pour une page convertie :

- Son contenu textuel issu des traitements de reconnaissance.
- Les coordonnées de chaque bloc de segmentation d'une image : de l'élément le plus grand qui est la page, découpée en espace imprimé avec ses marges et ses différents types de blocs, jusqu'à l'élément le plus petit qui est le mot ou l'espace (entre 2 mots). Ainsi chaque élément est défini par sa position dans l'image convertie.
- La note de confiance de chaque mot indiquant le niveau de certitude du moteur OCR pour sa reconnaissance.

Chaque fichier ALTO d'un document numérique doit contenir l'identifiant de la version image, par exemple « alto.NNNNNN » où NNNNNN est l'identifiant du

1. <http://www.loc.gov/standards/alto/techcenter/structure.php>

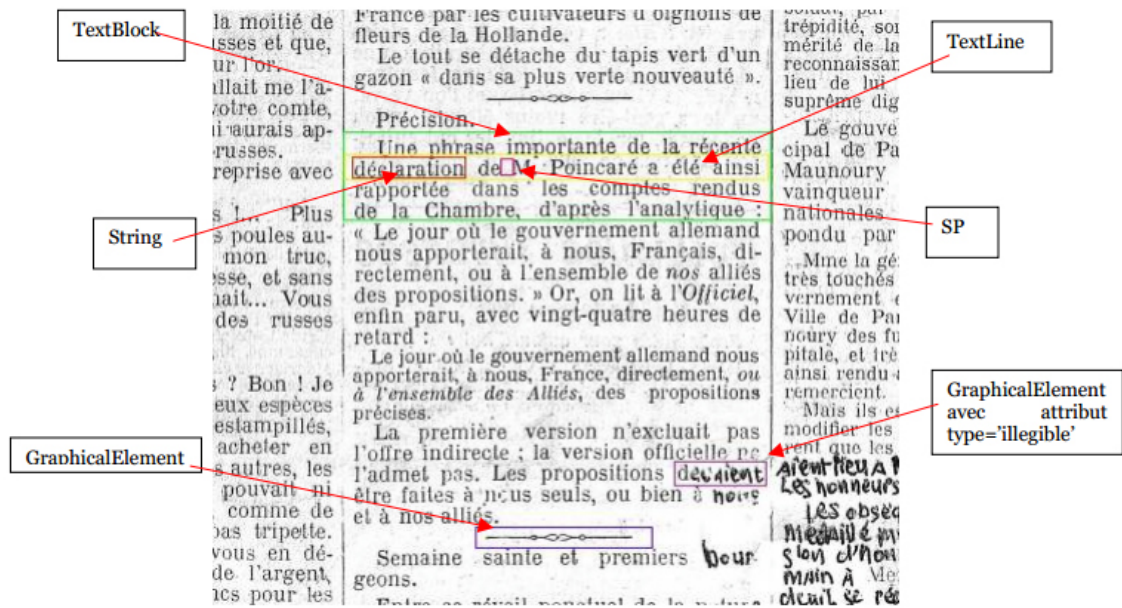


FIGURE 2.3 – Le traitement d'OCR

document numérisé. Un fichier ALTO se compose de trois sections principales que les enfants de l'élément racine `<alto>` :

- **La section `<Description>`** : Cette section méta-données sur le fichier ALTO. Elle permet de décrire la qualité de traitement de l'information de façon clair.
- **La section `<Styles>`** : Cette section contient les styles de texte et le paragraphe avec leurs descriptions individuelles :

✓ *La section `<TextStyle>`* : Cette balise est spécifique pour décrire le description de police du texte.

✓ *La section `<ParagraphStyle>`* : Cette balise est spécifique pour décrire les descriptions de paragraphe.

- **La section `<Layout>`** : Cette section contient les informations de contenu. Elle est subdivisée en éléments `<page>`.

✓ Une page se compose de marges et printspace, tous ceux sont non-intersection des zones rectangulaires au sein de la zone de la page. Chacun de ceux-ci peuvent contenir n'importe quel nombre d'objets comme des lignes, des images ou des zones de texte et plus. Le résultat de la segmentation doit permettre de faire correspondre le texte issu de la conversion à l'image par transparence grâce au calcul des coordonnées de la position des éléments dans l'image. On distingue l'encodage du texte lui-même de celui des marges, inutile lorsqu'elles sont vides de contenu :

★ **TopMargin** : est utilisé pour désigner la zone supérieure de la page du bord gauche au bord droit hors zone de texte. Quand c'est possible, il s'agit de la zone contenant le titre, l'ours, etc.

★ **BottomMargin** : est utilisé pour désigner la zone inférieure de la page du bord gauche au bord droit hors zone de texte.

★ **LeftMargin** : est utilisé pour désigner la zone gauche de la page hors zone supérieure, zone inférieure et zone de texte.

★ **RightMargin** : est utilisé pour désigner la zone droite de la page hors zone supérieure, zone inférieure et zone de texte.

Chacun de ces quatre éléments n'est renseigné que dans le cas où il contient au moins un élément **<BlockGroup>**.

★ **PrintSpace** : est utilisé pour désigner la zone de texte. Cet élément est obligatoire. Il contient au moins un élément **<BlockGroup>**.

L'ALTO distingue 3 niveaux de segmentation :

✓ *Le mot.*

✓ *La ligne de texte* : c'est un ensemble de mots sur une même ligne entre un début de colonne et une fin de colonne. Certains textes comme les gros titres peuvent s'étendre sur plusieurs colonnes.

✓ *Le bloc de texte* : c'est un ensemble de lignes formant un ensemble cohérent, par exemple le paragraphe. Dans le cas des césures de mots, il contient au moins les deux lignes concernées par la césure.

Les blocs sont découpés grâce aux quatre éléments de l'entité **BlockGroup** :

✓ *TextBlock* : permet de désigner le bloc de texte. Cet élément est utilisé pour regrouper les lignes de textes en un ensemble cohérent. Un **TextBlock** est divisé en **TextLines** et ceux-ci sont divisés en outre dans les chaînes et les espaces.

✓ *Illustration* : est utilisée pour désigner un élément graphique (dessin, photo ou schéma) en rapport avec le contenu d'un texte placé dans la même page. La zone concernée est généralement un rectangle pour lequel les attributs **Positions/Dimensions** suffisent. Pour les zones complexes, l'élément **Shape** peut être utilisé.

✓ *GraphicalElement* : est utilisé pour désigner un élément graphique (dessins, photos, schémas, publicités graphiques) sans rapport avec le contenu du texte. Cet élément peut aussi être utilisé pour décrire un élément de séparation intertextuel ou un élément textuel non reconnu en tant que tel par l'OCR (voir exemple ci-dessous).

✓ *ComposedBlock* : est utilisé pour permettre l'imbrication d'éléments **BlockGroup**.

5 Production des documents numériques à la BnF

Le processus de numérisation des documents à la BnF commence par une programmation annuelle des documents qui vont être numérisés. Les marchés de production des documents numériques commencent par l'envoi mensuel d'un BTA (Bordereau de traitement aller) au prestataire accompagné des fichiers physiques. Le prestataire vérifie la présence de tous les titres qui existent dans le bordereau de traitement mensuel. Puis il envoie un bordereau de réception qui contient les titres des documents bien reçus et les titres des documents manquants.

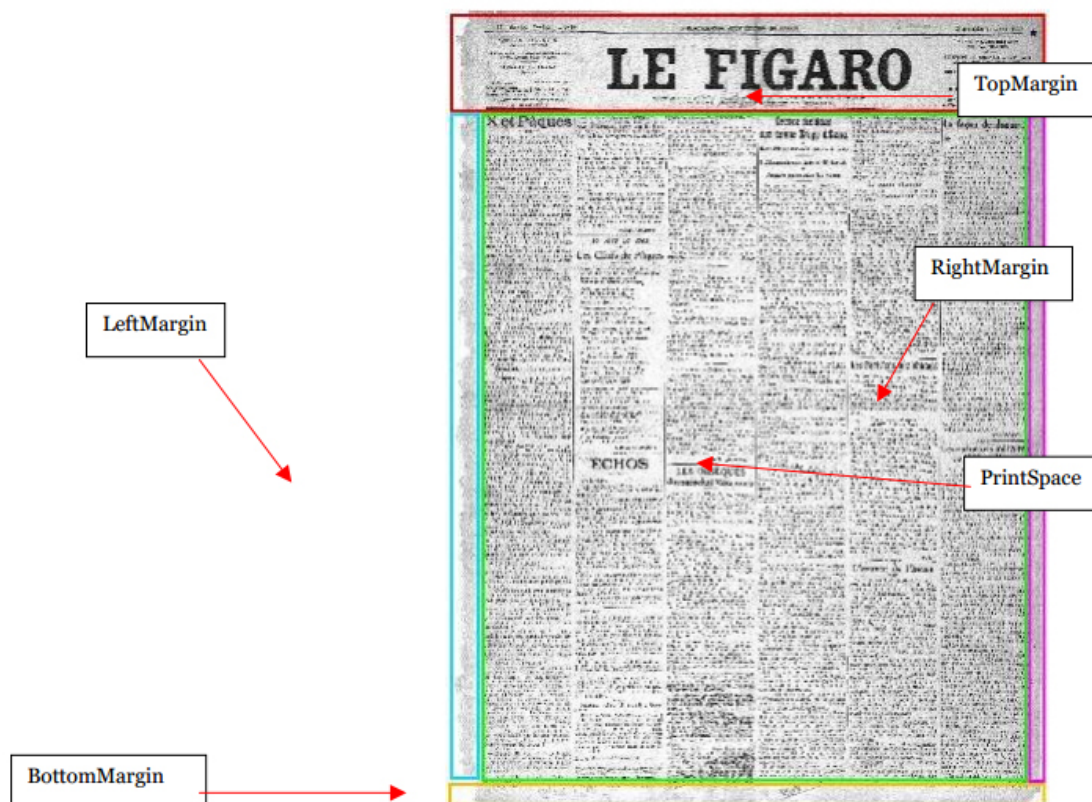


FIGURE 2.4 – Encodage des principaux éléments de la page

Pour chaque document numérisé, le prestataire prépare un répertoire numérique qui contient les images TIFF des pages du document, un fichier RefNum et un fichier TagTiff. Ces deux fichiers sont composés par des Métadonnées qui permettent de décrire les images et les documents originaux.

Le fichier RefNum englobe toutes les informations bibliographiques du document, les remarques des prestataires concernant les difficultés de numérisation et l'ordre des images du document. Un exemple de ce type de fichier est fourni dans l'annexe 1.

Le fichier TagTiff contient des informations qui décrivent la résolution de l'image, le taux de compression utilisé, l'espace de couleur de l'image, etc.... Le tableau 1 dans l'annexe 1 présente toutes les balises du document TagTiff qui sont utilisées dans la BnF.

Après la clôture du processus de numérisation, le prestataire d'acquisition prépare un BTR qui regroupe tous les titres et les identifiants des documents physiques numérisés. Puis il envoie ce bordereau à la BnF avec les documents physiques. Au même temps, L'ensemble des fichiers d'un document sont regroupés dans un répertoire de livraison faisant office de paquet de versement auquel le prestataire ajoute un fichier d'empreinte permettant de contrôler l'intégrité des données livrées, ce fichier est appelé BL (Bordereau de livraison des documents numériques). Tous les répertoires de livraison sont envoyés par voie électronique à travers un accès FTP

(protocole de transfert de fichiers) direct ou à travers des supports magnétiques (comme par exemple les disques durs).

Le département informatique prend en charge l'opération de l'intégration des documents numériques envoyés par les prestataires dans les serveurs internes de la Bibliothèque. Par ailleurs la chaîne d'entrée BnF et plus précisément le département informatique effectue un certain nombre de contrôles techniques faits sur les fichiers RefNum et TagTiff à travers des outils automatiques de vérification de structure physique. Ces examens cherchent à vérifier et valider la structure de ces fichiers. Si sa composition n'est pas conforme avec les normes des fichiers RefNum ou TagTiff, les documents numériques vont être rejetés partiellement.

En parallèle, le service numérisation assure un contrôle sur les documents numériques. Les examens appliqués cherchent à vérifier la conformité des informations bibliographiques qui se trouvent dans le fichier RefNum avec les données originaux du document. Le service valide également dans le cadre de ce contrôle les tableaux des indexes et l'ordre des images des pages du document. Les documents numériques qui contiennent des défauts sont rejetés partiellement.

Tous les documents numériques admis ou rejetés partiellement lors de ces contrôles sont revérifiés de nouveau par le même service. Les nouveaux examens de service numérisation vérifient d'une part les défauts détectés par le département informatique de la BnF, pour décider si les documents rejetés partiellement doivent être renvoyés au prestataire ou non.

D'autre part une inspection est assurée sur la conformité des propriétés de l'image avec les caractéristiques qui paraissent dans le cahier de charges et dans la charte de traitement.

Les erreurs détectées par le service numérisation sont rangées en deux catégories :

- Les erreurs mineures.
- Les erreurs majeures.

Les tableaux 2 et 3 présentés dans l'annexe 1 présentent les erreurs majeures et mineures. Tous les défauts qui existent dans les deux tableaux avec un fond coloré peuvent être aussi des sources de dégradation de taux d'OCR. La figure 3.1 montre les différentes phases de réalisation des marchés de numérisation ainsi que les échanges qui se font entre la BnF et le prestataire.

Après l'acquisition des images du document et la validation de la qualité de numérisation des documents, les prestataires de l'OCR se chargent de la conversion des images du document au texte numérique. Ce module de conversion est réalisé à l'aide des systèmes de l'OCR. Lorsqu'on doute signalé par le moteur de l'OCR sur la nature ou/et l'identité des formes qui existent sur l'image, la BnF exige une intervention humaine pour lever l'ambiguïté afin de respecter le niveau de qualité ciblée.

A la fin de l'opération de l'ocrisation, la BnF effectue trois genres de contrôle :

⇒ **Contrôle de conformité du fichier XML avec les spécifications de la BnF** : La BnF contrôle l'emplacement des blocs du fichier ALTO. En effet, la superposition des blocs de même genre n'est pas permise dans les spécifications de production des documents numériques.

⇒ **Contrôle de segmentation** : Ce contrôle vise à vérifier toutes les boîtes de segmentation en doute et la vérification des pages segmentées ainsi que le sens de l'écriture.

⇒ **Contrôle de reconnaissance** : Ce contrôle vise à repérer les erreurs de reconnaissance des mots selon la qualité de l'OCR spécifiée dans la cahier de charge.

A la fin de chaque opération de contrôle un rapport de contrôle est rédigé. Selon le types et le nombre des erreurs, la BnF choisit soit de renvoyer le document au prestataire pour qu'il soit corrigé, soit valider le document dans Gallica.

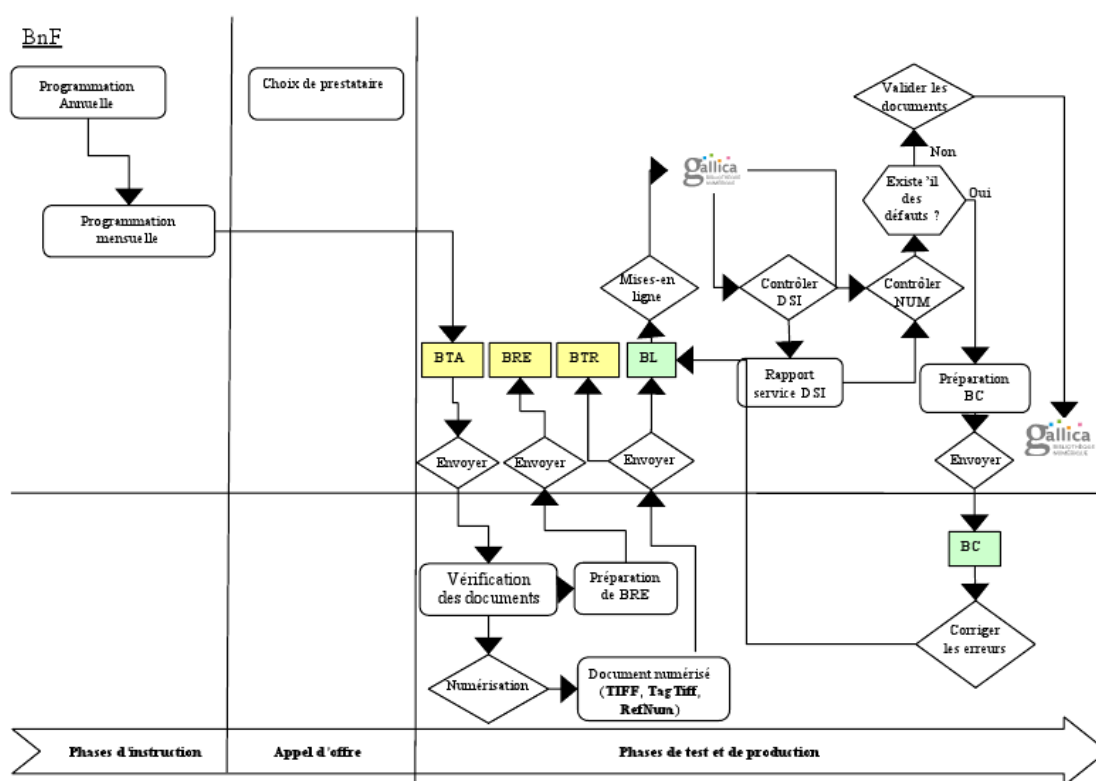


FIGURE 2.5 – Schéma de réalisation des marchés numérisation

5.1 Technique de lecture du fichier ALTO

Les fichiers XML sont traités avec deux techniques DOM et SAX. Dans mon application nous avons utilisé la technique DOM puisque la nature d'utilisation est riche de fonctionnalités et les objets construits à partir des fichiers XML persistent dans la mémoire.

C'est pourquoi on a utilisé la technique DOM pour la lecture et l'écriture des fichiers XML.

6 Conclusion

Dans ce chapitre on a défini quelque définition sur le domaine de travail et les techniques que nous avons utilisé pour réaliser le projet. Dans le troisième chapitre on va présenter la démarche préliminaire au travail et l'étude menée pour la modélisation de l'expression des besoins et la méthodologies utilisées dans la réalisation du projet.

Chapitre 3

Conception et Gestion de projet

1 Introduction

Le chapitre précédent se limite à une vue générale de l'application. Toutefois, ce chapitre présente en détail l'étude menée par une modification du domaine de l'application.

L'expression des besoins est la première étape qui fait apparaître les acteurs interagissant avec le système ainsi que ses entités internes. Ces besoins sont structurés par la suite dans le modèle d'analyse permettant de mieux comprendre le comportement des systèmes et ses limites.

2 Gestion de projet

Les méthodes agiles sont des groupes de pratiques pouvant s'appliquer à divers types de projets. Les méthodes agiles impliquent au maximum le demandeur (client) et permettent une grande réactivité à ses demandes. Elles visent la satisfaction réelle du besoin du client en priorité aux termes d'un contrat de développement.

Une méthode agile est donc avant tout itérative sur la base d'un affinement du besoin mis en œuvre dans des fonctionnalités en cours de réalisation et même déjà réalisées. Cet affinement, indispensable à la mise en œuvre du concept adaptatif, se réalise en matière de génie logiciel sous deux aspects :

- fonctionnellement, par adaptation systématique du produit aux changements du besoin détecté par l'utilisateur lors de la conception-réalisation du produit (notion de validation permanente de l'utilisateur avec RAD et notion de conception émergente avec XP).
- techniquement, par remaniement régulier du code déjà produit.

Les méthodes Agiles prônent 4 valeurs fondamentales :

- **L'équipe (« Les individus et leurs interactions plus que les processus et les outils »)** : Dans l'optique agile, l'équipe est bien plus importante que les outils (structurants ou de contrôle) ou les procédures de fonctionnement. Il est

préférable d'avoir une équipe soudée et qui communique, composée de développeurs (éventuellement à niveaux variables), plutôt qu'une équipe composée d'experts fonctionnant chacun de manière isolée. La communication est une notion fondamentale.

- **L'application (« Des logiciels opérationnels plus qu'une documentation exhaustive ») :** Il est vital que l'application fonctionne. Le reste, et notamment la documentation technique, est une aide précieuse mais non un but en soi. Une documentation précise est utile comme moyen de communication. La documentation représente une charge de travail importante, mais peut pourtant être néfaste si elle n'est pas à jour. Il est préférable de commenter abondamment le code lui-même, et surtout de transférer les compétences au sein de l'équipe (on en revient à l'importance de la communication).

• **La collaboration (« La collaboration avec les clients plus que la négociation contractuelle »)** : Le client doit être impliqué dans le développement. On ne peut se contenter de négocier un contrat au début du projet, puis de négliger les demandes du client. Le client doit collaborer avec l'équipe et fournir un feed-back continu sur l'adaptation du logiciel à ses attentes.

• **L'acceptation du changement (« L'adaptation au changement plus que le suivi d'un plan »)** : La planification initiale et la structure du logiciel doivent être flexibles afin de permettre l'évolution de la demande du client tout au long du projet. Les premières livraisons du logiciel vont souvent provoquer des demandes d'évolution.

Notre projet a démarré par la détermination des besoins de la BnF à travers les réunions que nous avons faites avec les responsables de pôle contrôle de la BnF. Ensuite, nous avons validé la conception de l'application et les scénarios de l'utilisation de l'application avec les encadreurs. La réalisation de notre application est faite selon plusieurs versions, chaque version est livrée à la BnF et nous recevons un rapport en contre partie qui décrit les erreurs et les modifications nécessaires de la part du service de numérisation. La première version par exemple permet d'afficher les composantes de l'image du documents.

Dans la deuxième version, nous avons proposé une interface capable d'inclure les algorithmes de contrôle automatique a côté image et cote répertoire d'images et après une version qui permet de faire comparaison entre deux OCR aussi cote image et cote répertoire d'images. A chaque version réalisé nous avons envoyé à la BnF pour vous rendre un rapport contient toutes les informations nécessaires. Dans les phases de développement de l'application nous avons adapté ce mécanisme de développement.

• **Planning prévisionnel**

Il permet de représenter graphiquement l'avancement du projet, permettant de visualiser les diverses tâches liées composant un projet.

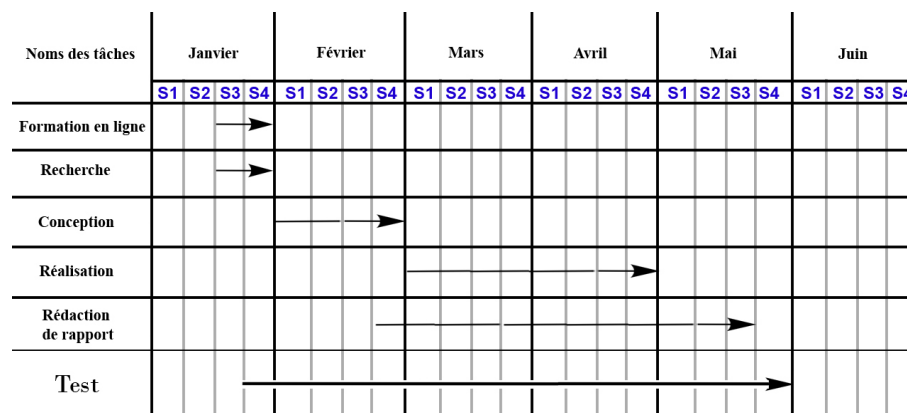


FIGURE 3.1 – Planning prévisionnel

3 Expression des besoins et modele d'analyse

3.1 Langage et modélisation unifiée

UML (en anglais Unified Modeling Language ou « langage de modélisation unifiée ») est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel.

UML est l'accomplissement de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard défini par l'Object Management Group (OMG).

UML 2.3 propose 13 types de diagrammes (9 en UML 1.3). UML n'étant pas une méthode, leur utilisation est laissée à l'appréciation de chacun, même si le diagramme de classes est généralement considéré comme l'élément central d'UML ; des méthodologies, telles que l'UnifiedProcess, axent elles l'analyse en tout premier lieu sur les diagrammes de cas d'utilisation (Use Case). De même, on peut se contenter de modéliser seulement partiellement un système, par exemple certaines parties critiques.

3.2 Pourquoi on a choisi l'UML comme une méthode de conception ?

L'approche objet est la dernière proposition dans la conception des systèmes d'information. Elle permet d'intégrer dans l'object des données et des traitements et prendre en compte une plus large gamme d'applications tout en favorisant la conception et réutilisation des compositions dont le but est d'améliorer d'une part la productivité et la rentabilité des concepteurs / développeurs et réduire d'autre part le coût de revient des applications.

C'est pourquoi nous avons adapté la méthode d'UML dans la conception de notre application.

3.3 Identification des acteurs

Dans le cas de ce projet , l'analyse de l'existant représente une source d'inspiration permettant de dégager les acteurs interagissant avec l'application à savoir :

Le contrôleur de la BNF : C'est l'utilisateur principal. Cette application lui permet de :

- Lancer le contrôle Automatique côté Image ou côté répertoire d'images.
- Lancer la Comparaison entre les deux OCR côté image ou côté répertoire d'images.
- Lancer une recherche de mots.
- corriger la segmentation.
- corriger les strings.

3.4 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un diagramme de cas d'utilisation est un graphe d'acteurs, un ensemble de cas d'utilisation englobés par la limite du système, des relations (ou associations) de communication (participation) entre les acteurs et les cas d'utilisation, et des généralisations de ces cas d'utilisation.

Les relations de généralisation/spécialisation entre acteurs permettent de définir des profils d'acteurs. Les relations entre cas d'utilisation sont soit utilisées (uses) pour éviter de dupliquer des processus communs à plusieurs processus, soit étend (extends) pour décrire séparément des parties alternatives ou optionnelles ou exceptionnelles de processus. Les relations entre acteurs et cas d'utilisation indiquent les interactions.

3.5 Diagramme de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language.

On montre ces interactions dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroule les actions entre les acteurs ou objets.

Les périodes d'activité des objets sont symbolisées par des rectangles. Plusieurs types de messages (actions) peuvent transiter entre les acteurs et objets :

- **message simple** : le message n'a pas de spécificité particulière d'envoi et de réception.
- **message avec durée de vie** : l'expéditeur attend une réponse du récepteur pendant un certain temps et reprend ses activités si aucune réponse n'a lieu dans un délai prévu.
- **message synchrone** : l'expéditeur est bloqué jusqu'au signal de prise en compte par le destinataire. Les messages synchrones sont symbolisés par des flèches barrées.
- **message asynchrone** : le message est envoyé, l'expéditeur continue son activité que le message soit parvenu ou pris en compte ou non. Les messages asynchrones sont symbolisés par des demi-flèches.
- **message dérobant** : le message est mis en attente dans une liste d'attente de traitement chez le récepteur.

4 Spécification des besoins

La spécification des besoins passe par l'énumération des besoins. les diagrammes de cas d'utilisation et de séquence nous seront utiles pour une spécification explicite

des besoins. Puisque nous avons travaillé avec la méthode Agile, nous avons décomposé ces diagrammes suivant en des scénarios déterminés à partir de diagrammes de cas d'utilisation général.

4.1 Diagramme de Cas d'utilisation général de l'application

L'utilisateur de l'application est l'utilisateur principal auquel s'adresse cette application. Il peut accéder aux informations décrivant le contenu des données du fichier ALTO telles que le contenu du Layout. Il peut également faire un contrôle de données du fichier ALTO à propos l'image analysée selon un contrôle de strings et un contrôle du segmentation et lancement du recherche. Il a le droit de modifier, supprimer et ajouter à chaque type de contrôle accédé. L'utilisateur peut fait une comparaison entre deux OCR soit cote image ou cote répertoires d'images. Il peut consulter l'aide de l'application ou l'aide de QT.(voir Figure 3.2)

FIGURE 3.2 – Diagramme de Cas d'utilisation général de l'application

aaa

Selon le diagramme du cas d'utilisation général les contrôleurs de la BnF peuvent corriger la reconnaissance et les coordonnées des éléments du fichier ALTO. Les 7 scénarios suivants décrivent les fonctionnalités de correction que notre interface doit assurer pour corriger les éléments des fichiers ALTOs :

✚ **Scénario 1** : L'utilisateur de l'application peut utiliser l'interface contrôle automatique pour lancer les algorithmes de vérification automatiques des éléments détectés par l'OCR dans le fichier ALTO et il peut afficher aussi les composants de l'image.

✚ **Scénario 2** : A travers l'interface **contrôle automatique de reconnaissance de chaîne de caractères**, l'utilisateur peut ajouter des chaînes de caractères qui ne sont pas reconnus par l'OCR.

✚ **Scénario 3** : L'utilisateur de l'application peut supprimer les fausses reconnaissances des mots dans le fichier ALTO à travers l'interface contrôle automatique de reconnaissance de chaîne de caractères.

✚ **Scénario 4** : L'utilisateur de l'application peut également modifier les mots qui se trouvent dans le fichier ALTO à travers l'interface de contrôle automatique de reconnaissance.

✚ **Scénario 5** : L'utilisateur peut modifier aussi la segmentation des différents composants du fichier ALTO (mots, Illustrations, les paragraphes, les lignes) à travers l'interface de **contrôle automatique de segmentation**.

✚ **Scénario 6** : L'utilisateur de l'application peut supprimer les éléments supplémentaires qui existent dans le fichier ALTO grâce à l'interface contrôle automatique.

✚ **Scénario 7** : L'utilisateur peut utiliser l'interface de contrôle automatique pour segmenter les éléments oubliés par l'OCR (les paragraphes, les lignes, les mots et les Illustrations).

Tous les scénarios de contrôle de reconnaissance et de segmentation des éléments ALTO qui sont applicables à travers l'interface de contrôle automatique sont également applicables dans l'interface de comparaison entre deux fichiers ALTO. A travers l'interface que nous allons développer, les contrôleurs de la BnF sont capables d'effectuer d'autres tâches comme la recherche visuelle des mots et la comparaison visuelle entre les résultats de segmentation de deux fichiers ALTO. Les scénarios qui décrivent ces fonctionnalités sont les suivants :

✚ **Scénario 8** : L'utilisateur de l'application peut lancer une recherche de mots qui dans le fichier ALTO.

✚ **Scénario 9** : L'utilisateur de l'application peut comparer les coordonnées des éléments de deux fichiers ALTO différents.

4.2 Gestion des scénarios

Les opérations d’affichage des éléments de fichier ALTO ou de contrôle de scripts des mots ou/et de segmentation des éléments sont faites à l’aide de deux types de projets **Contrôle automatique** et **Comparaison entre deux fichiers ALTO**. L’utilisateur de l’application peut contrôler la conversion soit d’une seule image ou d’un répertoire d’images (document). Nous allons montrer dans ce rapport 7 scénarios (1, 2, 3, 4, 5, 6 et 8) qui se réalise dans le cadre d’un projet de contrôle automatique de la conversion d’une image. Ensuite, nous allons voir un scénario (7) qui se réalise dans le cadre d’un projet de contrôle automatique de la conversion d’un répertoire d’images. A la fin de cette partie, nous découvrons le scénario 9 qui montre les fonctionnalités des projets de comparaison entre les fichiers ALTO qui se trouvent dans deux répertoires d’images différents. Les fichiers ALTO doivent être produits par deux OCRs différents.

① Création d’un projet de contrôle automatique de l’image

L’opération de contrôle automatique des fichiers ALTO commence par la création d’un projet de contrôle automatique des éléments d’un fichier ALTO. Pour cela, le système doit offrir à l’utilisateur la possibilité de choisir l’image et le fichier ALTO correspondant. Ensuite, l’interface doit contrôler la conformité du nom de l’image et du nom du fichier ALTO. Si les deux noms sont conformes, notre système doit afficher les composants de fichier ALTO (les paragraphes, les lignes, les mots et les illustrations) sur l’image sélectionnée par l’utilisateur de l’application.

Les étapes d’ouverture d’un projet de contrôle automatique sont les suivantes :

- ❶ Le système demande à l’utilisateur de choisir le chemin de l’image.
- ❷ Le contrôleur choisit le chemin de l’image.
- ❸ Le système vérifie l’existence de l’image.
- ❹ Le système accepte l’image.
- ❺ Le système demande à l’utilisateur de choisir le chemin du fichier ALTO correspondant à l’image introduite.
- ❻ L’utilisateur choisit le chemin du fichier ALTO.
- ❼ Le système vérifie l’existence du fichier ALTO et la conformité du nom du fichier ALTO avec le nom de l’image choisie.
- ❽ A la fin de cette opération, L’interface affiche l’image et les composants du fichier ALTO.

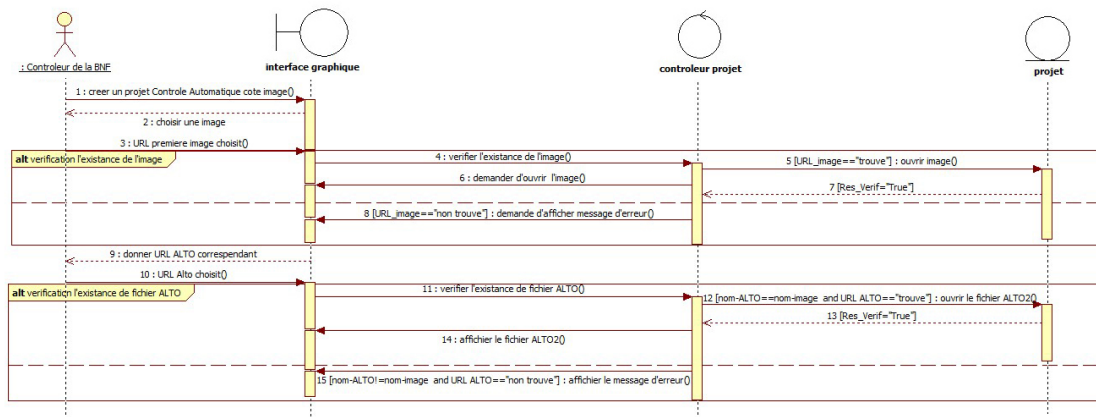


FIGURE 3.3 – Diagramme de séquence (creation projet contrôle automatique côté image)

❖ Scénario 1

L'utilisateur de l'application peut utiliser l'interface contrôle automatique pour lancer les algorithmes des vérifications automatiques des éléments du fichier ALTO et il peut également afficher des composants de l'image.

☆ *Diagramme de cas d'utilisation*

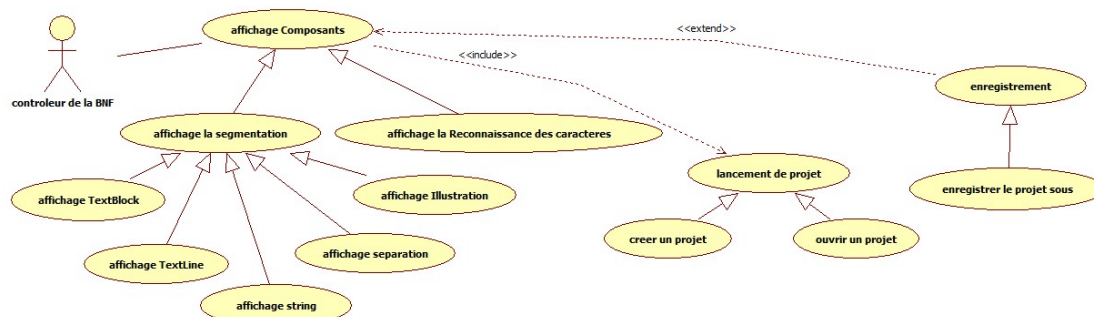


FIGURE 3.4 – Diagramme de cas d'utilisation scénario1

□ DESCRIPTION DU SCÉNARIO :

- **Nom du cas d'utilisation :** L'affichage des composants de fichier ALTO.
- **Acteur Principal :** Les contrôleurs de la BNF.
- **Objectif :**
 - Afficher les boîtes englobant des éléments du fichier ALTO.
 - consulter le contenu des mots qui existent dans le fichier ALTO.
- **Pré-condition :** L'acteur principal doit accéder aux outils d'affichage d'éléments du fichier ALTO.

- **Post-condition** : Les boîtes englobant des éléments sont affichées.

□ SCÉNARIO DE TEST NOMINAL :

Après la création du projet de contrôle automatique, le contrôleur peut afficher des éléments textuelles comme les mots, les phrases et les paragraphes et des éléments graphiques comme les illustrations. Les étapes d’affichage sont les suivantes :

- ❶ L’utilisateur sélectionne le bouton de l’élément à afficher
- ❷ L’interface doit afficher toutes les boîtes englobant qui appartient au type d’élément sélectionné

Le contrôleur de la BnF peut aussi consulter les mots qui existent dans le fichier ALTO. Cette opération est réalisée grâce à l’éditeur des mots qui affiche la ligne encore de modification sur l’image de la page choisie dans l’étape de création de projet. La consultation des mots se fait à travers les étapes suivantes :

- ❶ L’utilisateur peut choisir l’option de correction des mots.
- ❷ Le système doit afficher tous les mots sous forme d’un tableau qui contient toutes les phrases du fichier ALTO dans ses lignes.
- ❸ L’utilisateur peut sélectionner les phrases qui contiennent les mots à corriger pour modifier les mots du fichier ALTO.
- ❹ Le système doit dessiner la boîte englobant la phrase en cours de modification sur l’image pour permettre à l’utilisateur de se positionner sur l’image.
- ❺ L’utilisateur peut demander l’enregistrement de son projet de correction automatique.
- ❻ Le système doit enregistrer le projet et modifier les éléments du fichier ALTO.

□ SCÉNARIO DE TEST ALTERNATIF :

Notre système doit être capable de gérer les traitements incorrects ou interdits effectués par l’utilisateur de l’application. Le processus de gestion de ce genre d’erreurs est réalisé à travers des messages qui apparaissent dans des fenêtres d’information à chaque traitement incorrect. Dans le cas de l’ouverture du projet de contrôle automatique de la conversion d’une image, si le nom de l’image n’est pas conforme au nom de fichier ALTO notre système doit générer un message d’erreur. Cette opération est réalisée à travers les étapes suivantes :

- ❶ L’utilisateur choisit un fichier ALTO ayant un nom différent du nom de l’image sélectionnée au début de l’opération de l’ouverture de projet.
- ❷ Le système doit indiquer un message d’erreur pour informer l’utilisateur que le nom de l’image et le nom fichier ALTO sont différents.

☆ *Diagramme de Séquence* :

□ DESCRIPTION DU DIAGRAMME :

Le diagramme de séquence ci-dessus illustre l’opération d’affichage des composants de fichier ALTO. L’usager de l’application commence par créer un projet de

contrôle automatique de l'image. Ensuite, l'utilisateur sélectionne le genre d'éléments à afficher sur l'image de la page. Pour consulter les mots qui existent dans le fichier ALTO, l'utilisateur choisit l'option de la correction des mots pour qu'il consulte tous les mots du fichier ALTO dans un éditeur qui sépare les phrases qui se trouvent dans le fichier ALTO.

Les éléments que l'utilisateur peut consulter sont les mots, les phrases, les paragraphes, les espaces entre mots et les illustrations. Cette opération d'affichage commence par la sélection du bouton qui correspond à l'élément que l'utilisateur veut afficher. Une fois le bouton est sélectionné, le contrôleur des requêtes consulte la liste des éléments à afficher pour récupérer les coordonnées de ses boîtes englobantes. Après la récupération des coordonnées des éléments, l'interface trace les boîtes englobantes des éléments demandés par l'utilisateur.

L'opération de la consultation de contenu des mots commence par la sélection de l'option contrôle de la reconnaissance des strings. Une fois cette option est sélectionnée le système consulte toutes les phrases qui existent dans le fichier ALTO pour créer l'éditeur des mots qui contient en lignes toutes les lignes de fichier ALTO. Ensuite, le système remplit chaque ligne de l'éditeur par les mots qui existent dans la phrase correspondant. A la fin de cette opération, le système affiche l'éditeur des mots sur l'interface de l'application.

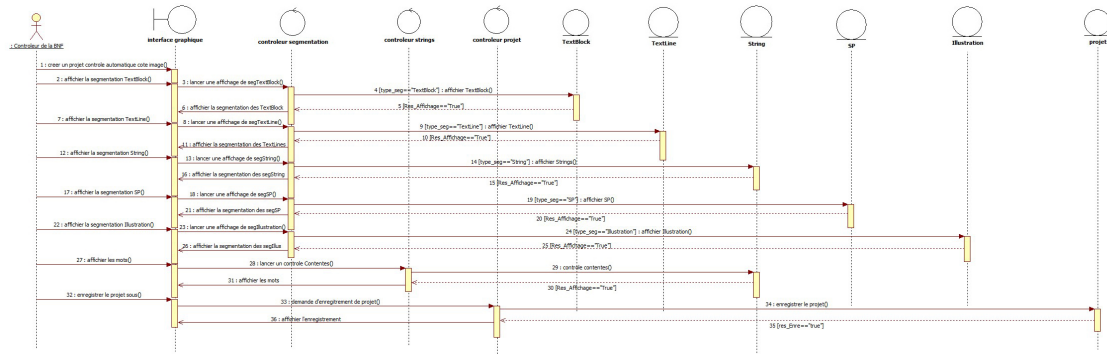


FIGURE 3.5 – Diagramme de Séquence scénario1

aaa

❖ Scénario 2

A travers l'interface **contrôle automatique de reconnaissance de chaîne de caractère**, l'utilisateur peut ajouter des chaînes de caractère qui ne sont pas reconnus par l'OCR :

☆ Diagramme de cas d'utilisation

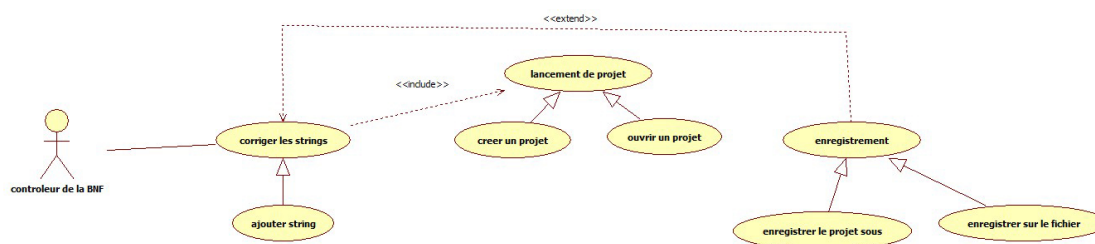


FIGURE 3.6 – Diagramme de cas d'utilisation scénario2

□ DESCRIPTION DU SCÉNARIO :

- **Nom du cas d'utilisation** : Ajout des mots dans l'opération de correction manuelle des mots.
- **Acteur Principal** : Les contrôleurs de la Bibliothèque nationale de France.
- **Objectif** :
 - Ajouter des mots.
- **Pré-condition** : L'acteur principal doit accéder à l'outil de correction des mots.
- **Post-condition** : Opération d'ajout des mots réussite.

□ SCÉNARIO DE TEST NOMINAL :

Après la création du projet de contrôle automatique de la conversion, l'utilisateur peut contrôler et corriger les erreurs de la reconnaissance des mots à travers l'éditeur des mots que notre application doit fournir à l'utilisateur. Dans notre cas, l'usager a la possibilité d'ajouter des mots qui ne sont pas détecté par l'OCR. Le scénario nominale de déroulement de l'opération de l'ajout des mots est le suivant :

- ❶ L'utilisateur choisi l'option de correction des strings.
- ❷ Le système doit afficher tous les mots de fichier ALTO dans l'éditeur des chaînes de caractère.
- ❸ L'utilisateur peut choisir la phrase qui contient des mots manquants afin d'ajouter les éléments manqués.
- ❹ La boîte englobant de chaque phrase encoure de modification (Ajout) est dessiné sur l'image de la page.
- ❺ L'utilisateur peut ajouter les mots manqués.
- ❻ Durant l'opération d'ajout des mots manqués, l'utilisateur peut enregistrer les modifications réalisés sur le fichier ALTO.

⑦ A chaque demande d'enregistrement, le système doit mettre à jour le contenu du fichier ALTO.

☆ *Diagramme de Séquence :*

□ DESCRIPTION DU DIAGRAMME :

Le diagramme de séquence ci-dessus illustre le scénario de mécanisme d'ajout des mots dans l'éditeur de notre application. A travers le diagramme de séquence et comme toute opération de modification de contenu de fichier ALTO, l'utilisateur de l'application commence par la création d'un projet de contrôle automatique de l'image. Ensuite, le contrôleur de la BNF sélectionne l'option contrôler strings qui va générer l'éditeur des mots sous forme d'un tableau. Les lignes de ce tableau représentent toutes les phrases du fichier ALTO.

Pour ajouter les mots manqués, l'utilisateur doit se positionner sur l'image du document pour faciliter l'opération de vérification des mots trouvés par l'OCR. Si il y'a des mots manquants dans la ligne courante alors l'utilisateur peut ajouter les mots oubliés.

Durant l'opération de l'ajout des mots, l'utilisateur peut demander l'enregistrement des modifications qu'il a réalisé. A ce moment, le système doit mettre à jour le contenu du fichier ALTO.

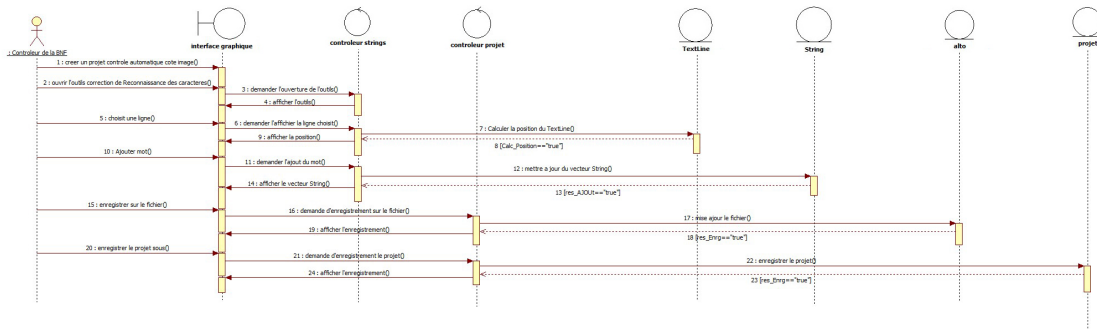


FIGURE 3.7 – Diagramme de séquence scénario2

❖ Scénario 3

L'utilisateur de l'application peut supprimer les fausses reconnaissance des mots dans le fichier ALTO à travers l'interface contrôle automatique de reconnaissance de chaîne de caractère.

☆ Diagramme de cas d'utilisation

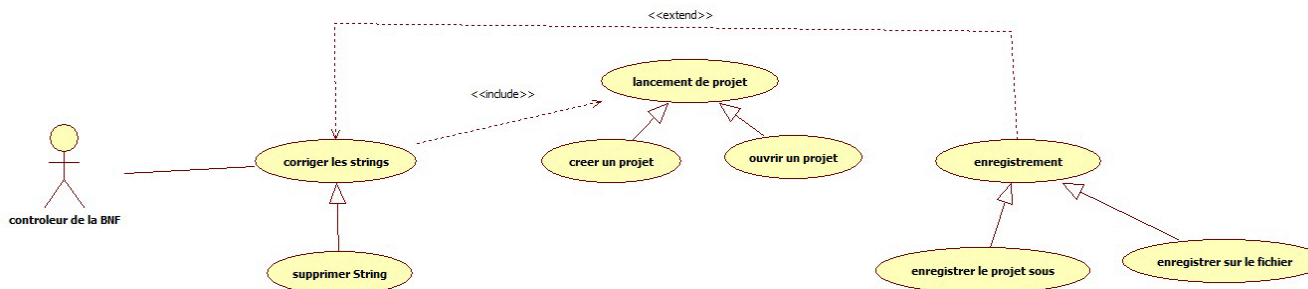


FIGURE 3.8 – Diagramme de cas d'utilisation scénario3

❑ DESCRIPTION DU SCÉNARIO :

- **Nom du cas d'utilisation :** Suppression d'un mot ou d'une partie de mot.
- **Acteur Principal :** Les contrôleurs de la Bibliothèque nationale de France.
- **Objectif :**
 - Supprimer des mots incorrecte.
 - Supprimer la partie incorrecte d'un mot.
- **Pré-condition :** Les contrôleurs de la Bibliothèque nationale de France doit accéder à l'outil de correction des strings (Editeur textuel spécial).
- **Post-condition :** Opération de supprimer des mots incorrecte est réussite.

❑ SCÉNARIO DE TEST NOMINAL :

Après la création du projet de contrôle automatique, le contrôleur doit avoir la possibilité de contrôler et corriger l'existence des mots incorrects dans le fichier ALTO. Cette opération doit être réalisée à travers un éditeur textuel spécial qui permet d'afficher la phrase qui est encore de modification sur l'image de la page. Cette fonctionnalité a pour objectif de faciliter l'opération de correction des mots. L'utilisateur de l'application peut également enregistrer ses modifications durant l'opération de correction de contenu du fichier ALTO. Le scénario nominale de déroulement de la suppression des mots ou d'une partie de mot est le suivante :

- ❶ Pour afficher l'éditeur des mots, l'utilisateur doit cliquer sur l'option correction des mots.
- ❷ Le système doit afficher tous les mots dans un éditeur tabulaire qui contient en ligne les phrases du fichier ALTO.
- ❸ Pour afficher la phrase non vérifiée sur l'image de la page, l'utilisateur doit sélectionné la ligne du tableau qui correspond à cette phrase.

- ④ Le système doit dessiner la boîte englobant de chaque phrase sélectionnée.
- ⑤ A travers l'éditeur textuel, l'utilisateur supprime le mot ou la partie de mot incorrecte.
- ⑥ Après la suppression des mots incorrects, l'utilisateur peut demander l'enregistrement de ses modifications.
- ⑦ Le système doit mettre à jour le contenu du fichier ALTO.

☆ *Diagramme de Séquence :*

□ DESCRIPTION DU DIAGRAMME :

Le diagramme de séquence ci-dessus illustre le scénario de mécanisme de suppression des mots dans notre application. Si le projet de contrôle automatique de l'image n'est pas créé, l'opération de suppression commence donc par la création du projet. Ensuite, l'utilisateur doit choisir l'outil de correction des mots afin d'afficher l'éditeur des mots qui se présente sous forme d'un tableau. Les lignes de notre éditeur tabulaire représentent les phrases qui existent dans le fichier ALTO. Après l'affichage de l'éditeur sur l'interface graphique, l'utilisateur procède à supprimer les mots incorrects et les fausses parties des mots.

L'utilisateur peut également demander l'enregistrement de ses modifications à travers l'interface graphique. Le contrôleur des mots **contrôleur strings** met à jour le vecteur des mots qui se trouvent dans la class métier **String**. Puis le contrôleur de lecture/écriture du fichier ALTO, procède à mettre à jour le contenu du fichier ALTO.

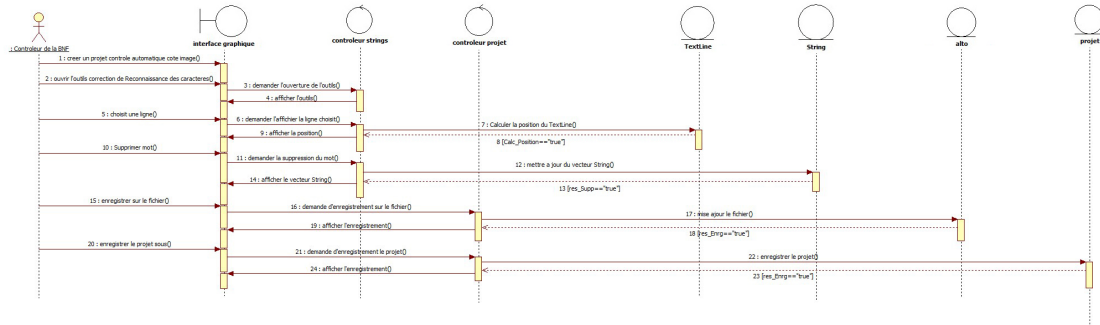


FIGURE 3.9 – Diagramme de séquence scénario3

❖ Scénario 4

L'utilisateur de l'application peut également modifier les mots qui se trouvent dans le fichier ALTO à travers l'interface de contrôle automatique de reconnaissance.

☆ Diagramme de cas d'utilisation

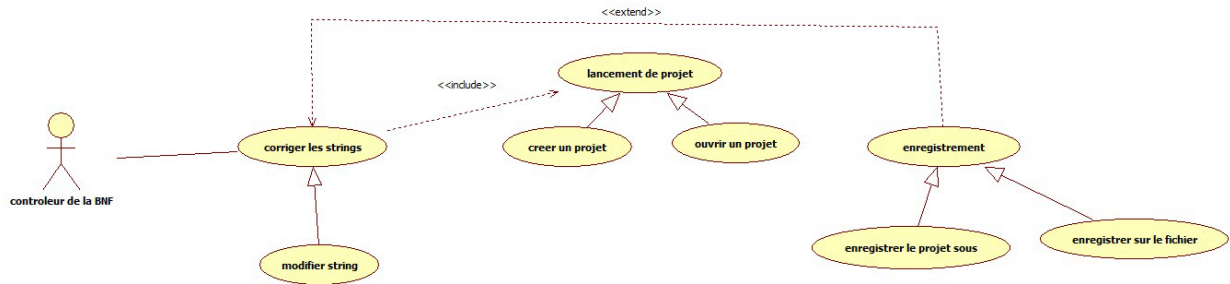


FIGURE 3.10 – Diagramme de cas d'utilisation scénario4

❑ DESCRIPTION DU SCÉNARIO :

- **Nom du cas d'utilisation** : Modifier des mots du fichier ALTO.
- **Acteur Principal** : Les contrôleurs de la Bibliothèque nationale de France.
- **Objectif** :
 - modifier les mots.
- **Pré-condition** : L'acteur principal doit accéder à l'outil de correction des mots (Editeur textuel).
- **Post-condition** : Opération de modification des mots réussite.

❑ SCÉNARIO DE TEST NOMINAL :

Après la création du projet de contrôle automatique, l'utilisateur peut corriger manuellement les erreurs des mots. D'après le diagramme de cas d'utilisation, à travers notre application le contrôleur a la possibilité de modifier les mots qui sont mal détectés par l'OCR. A la fin de l'opération de modification des mots, l'utilisateur peut enregistrer ces modifications.

Le scénario nominale de déroulement de l'opération de modification des mots est le suivant :

- ❶ L'utilisateur clique sur l'outil de correction des mots pour lancer l'éditeur textuel.
- ❷ Le système affiche tous les mots dans les mots du fichier ALTO dans l'éditeurs des mots.
- ❸ L'utilisateur sélectionne les phrases qui se trouvent dans les lignes de l'éditeur tabulaire afin de les modifier.
- ❹ Le système dessine la boîte englobant de chaque phrase sélectionnée sur l'image de la page.
- ❺ L'utilisateur vérifie et modifie les mots des phrases.

- ⑥ L'utilisateur peut demander l'enregistrement de ses modifications.
- ⑦ A chaque demande d'enregistrement, le système met à jour le contenu du fichier ALTO.

☆ *Diagramme de Séquence :*

□ DESCRIPTION DU DIAGRAMME :

Le diagramme de séquence ci-dessus illustre le scénario de mécanisme de la modification des mots. L'utilisateur de l'application commence par créer un projet de contrôle automatique de l'image. Ensuite, pour ouvrir l'éditeur tabulaire de texte, l'utilisateur choisit l'outil de correction des mots. Chaque ligne dans l'éditeur correspond à une phrase (TexteLine) dans le fichier ALTO.

L'utilisateur sélectionne les lignes de l'éditeur pour modifier et corriger les mots incorrects. Chaque ligne sélectionnée dans l'éditeur doit être affichée sur l'image de la page pour faciliter l'opération de vérification et de modification.

Le contrôleur de la BnF peut demander l'enregistrement des modifications réalisées sur les mots de l'éditeur et à ce moment, la class contrôleur **contrôleur strings** met à jour le vecteur des mots qui existe dans la class métier **String** et le contenu du fichier ALTO.

FIGURE 3.11 – Diagramme de séquence scénario4

❖ Scénario 5

L'utilisateur peut modifier aussi la segmentation les différents composants du fichier ALTO (mots, Illustrations, les paragraphes, les lignes) à travers l'interface de **contrôle automatique de segmentation**.

☆ Diagramme de cas d'utilisation

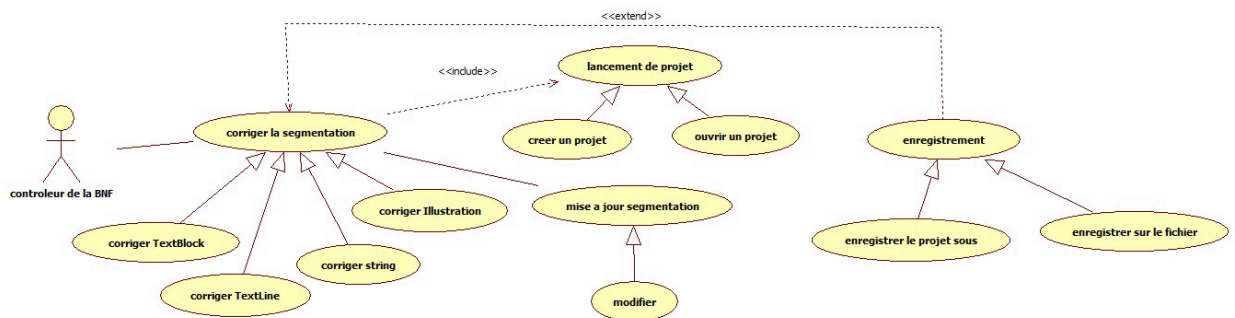


FIGURE 3.12 – Diagramme de cas d'utilisation scénario5

□ DESCRIPTION DU SCÉNARIO :

- **Nom du cas d'utilisation** : Modifier de la segmentation cote correction segmentation.
- **Acteur Principal** : Les contrôleurs de la Bibliothèque nationale de France.
- **Objectif** :
 - Modifier les coordonnées des éléments qui existent dans le fichier ALTO.
- **Pré-condition** : L'utilisateur de l'application doit accéder à l'interface de correction de la segmentation.
- **Post-condition** : Opération de modification de la segmentation réussite.

□ SCÉNARIO DE TEST NOMINAL :

L'opération de contrôle et de correction de segmentation commence après la création du projet de contrôle automatique. A travers les outils que notre application propose, L'utilisateur de l'application peut vérifier et corriger la segmentation des éléments du fichier ALTO à l'aide des outils de segmentation spécialisés. L'utilisateur de l'application peut enregistrer aussi les modifications qu'il a réalisé à n'importe quel instant de notre traitement.

Les étapes de la modification de segmentation sont suivantes :

- ❶ L'utilisateur clique sur l'outil de correction de la segmentation.
- ❷ Le système affiche un menu pour corriger et modifier les différents éléments qui peuvent exister dans un fichier ALTO.
- ❸ L'utilisateur procède à la modification des coordonnées des boîtes englobant (des paragraphes, des phrase, des Strings et des illustrations).
- ❹ Le système contrôle la mise à jour attribuer et lancer son processus de correction de la segmentation sur l'image.

- ⑤ L'utilisateur peut enregistrer les modifications qu'il a effectuées.
- ⑥ Le système doit mettre à jour le contenu du fichier ALTO.

☆ *Diagramme de Séquence :*

□ DESCRIPTION DU DIAGRAMME :

Le diagramme de séquence ci-dessus illustre le scénario de la modification des coordonnées des boîtes englobant des éléments qui existent dans le fichier ALTO. Ce scénario permet d'expliquer la modification de la segmentation des objets. Dans laquelle, l'utilisateur de l'application choisit l'objet (paragraphe, phrase, String ou Illustration) et la tâche de modification. Ensuite il a la possibilité de jouer sur l'image et modifie n'importe quel segmentation d'objet sélectionné.

L'utilisateur de l'application commence le processus de modification des coordonnées des boîtes englobant par le choix ou création d'un projet de contrôle automatique d'une image de document. Ensuite, le utilisateur choisit le type de l'élément qu'il va modifier ses coordonnées. Après, le contrôleur procède à la modification des coordonnées des éléments du fichier ALTO qui sont affichés sur l'interface principale de notre application. Chaque opération de modification stimule la classe "contrôleur segmentation" pour mettre à jour les coordonnées de l'élément qui a subi les modifications.

A chaque moment dans l'opération de correction des coordonnées du fichier ALTO, l'utilisateur peut enregistrer ses modifications.

FIGURE 3.13 – Diagramme de séquence scénario5

❖ Scénario 6

L'utilisateur de l'application peut supprimer les éléments supplémentaires qui existent dans le fichier ALTO grâce à l'interface contrôle automatique.

☆ Diagramme de cas d'utilisation

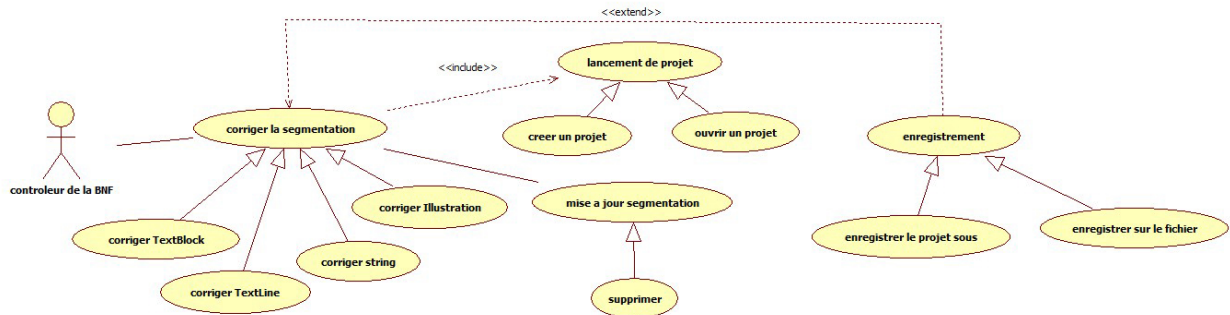


FIGURE 3.14 – Diagramme de cas d'utilisation scénario6

□ DESCRIPTION DU SCÉNARIO :

- **Nom du cas d'utilisation** : suppression des boîtes englobant des éléments incorrectes.

- **Acteur Principal** : Les contrôleurs de la Bibliothèque nationale de France.

- **Objectif** :

- Supprimer des éléments incorrects.

- **Pré-condition** : L'utilisateur de l'application doit accéder à l'outil de correction de la segmentation.

- **Post-condition** : Opération de suppression d'éléments incorrectes réussite.

□ SCÉNARIO DE TEST NOMINAL :

L'opération de contrôle et de correction de segmentation commence avec la création du projet de contrôle automatique. Puis à travers les outils fournis par notre système pour corriger les boîtes englobant des éléments du fichier ALTO, le contrôleur de la BnF peut supprimer les boîtes englobant incorrects. A la fin de l'opération de suppression des éléments incorrects du fichier ALTO, l'utilisateur peut enregistrer ses modifications.

Les étapes de la suppression de segmentation sont les suivantes :

- ➊ L'utilisateur clique sur l'outil de correction de la segmentation afin d'afficher les outils de vérification et de correction manuelle de segmentation.

- ➋ Le système affiche le menu des outils de correction segmentation.

- ➌ L'utilisateur supprime les boîtes englobant des éléments (paragraphe, phrase, mot, illustration) incorrects.

- ➍ Le système met à jour la liste des éléments traités à chaque opération de suppression.

- ➎ L'utilisateur demande l'enregistrement de ses modifications.

⑥ Le système met à jour le fichier ALTO.

☆ *Diagramme de Séquence :*

□ DESCRIPTION DU DIAGRAMME :

Le diagramme de séquence ci-dessus illustre le scénario de suppression des boîtes englobant des éléments incorrects. L'utilisateur de l'application commence par créer un projet contrôle automatique d'image. Ensuite, il choisit l'outil de la correction du segmentation afin d'afficher le menu de correction de segmentation.

Après l'utilisateur choisit le type de modification de fichier ALTO, dans notre cas nous allons supprimer des éléments, puis le genre des éléments que nous voulons les supprimés.

A chaque moment de l'opération de suppression, l'utilisateur peut enregistrer ses modifications.

FIGURE 3.15 – Diagramme de séquence scénario6

② Projet Contrôle Automatique cote répertoire d'images ‘

Comme nous avons présenté dans l'introduction de ce chapitre, notre application permet de contrôler un ensemble d'images qui sont localisées dans un seul répertoire. Cette fonctionnalité permet de vérifier et corriger des documents numériques et pas une image de la page.

La création d'un projet de contrôle de répertoire d'images commence par la sélection du répertoire d'images du document que nous voulons corriger ainsi que le répertoire des fichiers ALTO correspondants. Ensuite, après la vérification de la conformité des noms des images avec les noms des fichier ALTO dans les deux répertoires sélectionnés. Si tous les noms sont conformes, notre système affiche chaque image avec les composants de son fichier ALTO (les paragraphes, les lignes, les mots et les illustrations).

Les étapes d'ouverture de projet de contrôle automatique des répertoire d'images sont les suivants :

- ❶ L'utilisateur demande la création d'un projet de contrôle automatique d'un répertoire d'images.
- ❷ Le système demande à l'utilisateur de choisir le chemin du répertoire d'images.
- ❸ Le système vérifie l'existence des images dans le répertoire.
- ❹ Le système demande a l'utilisateur de choisir de le répertoire des fichiers ALTO correspondant à l'image introduite.
- ❺ Le système vérifie l'existence des fichiers ALTO dans le répertoire et la conformité des noms des images avec les noms des fichiers ALTO.
- ❻ Si chaque image du document numérique correspond à un fichier ALTO alors notre système affiche la première image de notre document numérique avec les éléments de son fichier ALTO.

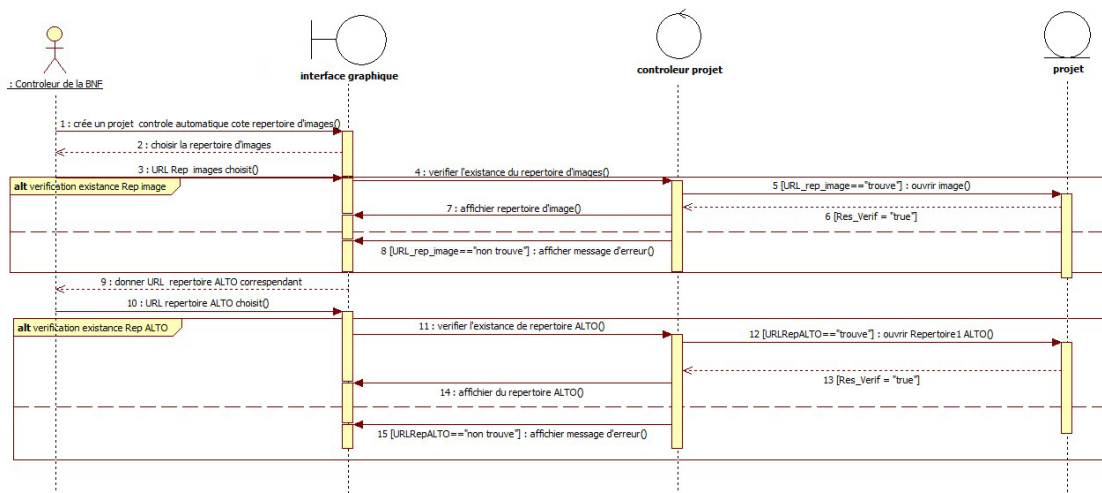


FIGURE 3.16 – Diagramme de séquence (creation de projet controle automatique cote répertoire d'images)

❖ Scénario 7

L'utilisateur de l'application peut supprimer les éléments supplémentaires qui existe dans le fichier ALTO grâce à l'interface contrôle automatique.

☆ Diagramme de cas d'utilisation

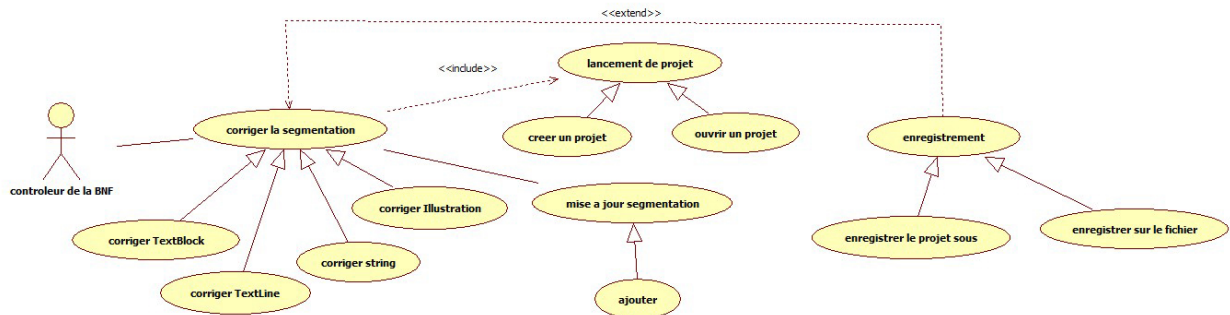


FIGURE 3.17 – Diagramme de cas d'utilisation scénario7

□ DESCRIPTION DU SCÉNARIO :

- **Nom du cas d'utilisation** : suppression des boîtes englobant des éléments incorrectes.

- **Acteur Principal** : Les contrôleurs de la Bibliothèque nationale de France.

- **Objectif** :

- Supprimer des éléments incorrects.

- **Pré-condition** : L'utilisateur de l'application doit accéder à l'outil de correction de la segmentation.

- **Post-condition** : Opération de suppression d'éléments incorrectes réussite.

□ SCÉNARIO DE TEST NOMINAL :

L'opération de contrôle et de correction de segmentation commence avec la création du projet de contrôle automatique. Puis à travers les outils fournis par notre système pour corriger les boîtes englobant des éléments du fichier ALTO, le contrôleur de la BnF peut supprimer les boîtes englobant incorrects. A la fin de l'opération de suppression des éléments incorrects du fichier ALTO, l'utilisateur peut enregistrer ses modifications.

Les étapes de la suppression de segmentation sont les suivantes :

- ➊ L'utilisateur clique sur l'outil de correction de la segmentation afin d'afficher les outils de vérification et de correction manuelle de segmentation.

- ➋ Le système affiche le menu des outils de correction segmentation.

- ➌ L'utilisateur supprime les boîtes englobant des éléments (paragraphe, phrase, mot, illustration) incorrects.

- ➍ Le système met à jour la liste des éléments traités à chaque opération de suppression.

- ➎ L'utilisateur demande l'enregistrement de ses modifications.

⑥ Le système met à jour le fichier ALTO.

☆ *Diagramme de Séquence :*

□ DESCRIPTION DU DIAGRAMME :

Le diagramme de séquence ci-dessus illustre le scénario de suppression des boîtes englobant des éléments incorrects. L'utilisateur de l'application commence par créer un projet contrôle automatique d'image. Ensuite, il choisit l'outil de la correction du segmentation afin d'afficher le menu de correction de segmentation.

Après l'utilisateur choisit le type de modification de fichier ALTO, dans notre cas nous allons supprimer des éléments, puis le genre des éléments que nous voulons les supprimés.

A chaque moment de l'opération de suppression, l'utilisateur peut enregistrer ses modifications.

FIGURE 3.18 – Diagramme de séquence scénario7

❖ Scénario 8

L'utilisateur de l'application peut lancer une recherche de mot dans le fichier ALTO.

☆ Diagramme de cas d'utilisation

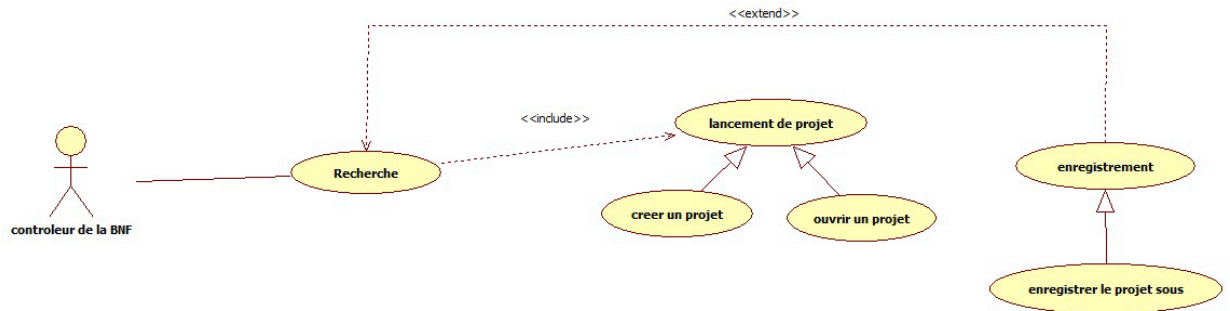


FIGURE 3.19 – Diagramme de cas d'utilisation scénario8

□ DESCRIPTION DU SCÉNARIO :

- **Nom du cas d'utilisation :** Recherche des mots dans le fichier ALTO.
- **Acteur Principal :** Les contrôleurs de la Bibliothèque nationale de France.
- **Objectif :**
 - Rechercher des mots dans le fichier ALTO.
- **Pré-condition :** L'utilisateur de l'application peut accéder à l'outil de recherche des mots.
- **Post-condition :** Opération de recherche des mots réussite.

□ SCÉNARIO DE TEST NOMINAL :

Dans ce scénario et après la création d'un projet automatique, le contrôleur peut effectuer une opération de recherche des mots dans le fichier ALTO à travers l'outil de recherche que notre application propose. Les étapes de recherche d'un mot sont :

- ❶ L'utilisateur peut afficher la fenêtre de recherche des mots en cliquant sur l'outil Recherche ou en utilisant le raccourci ctrl+F.
- ❷ Le système affiche une fenêtre de recherche de mot.
- ❸ L'utilisateur tape le mot à recherché
- ❹ Le système lance son algorithme de recherche sur le fichier ALTO et affiche les boîtes englobant des mots trouvés sur l'image du document traitée et le nombre des éléments retrouvés dans la fenêtre des résultats de recherche.

□ SCÉNARIO DE TEST ALTERNATIF :

Notre système doit être capable de gérer les traitements incorrects ou interdits effectués par l'utilisateur de l'application. Dans le cas que lorsque l'utilisateur n'est pas

entrée un chaîne a recherche. le systeme de gestion des erreurs affiche une message d'alerte.

Le scénario suivant indique ses etapes :

- ❶ L'utilisateur demande de chercher des mots.
- ❷ Le systeme l'onglet de recherche.
- ❸ L'utilisateur n'est pas entrée un chaîne a chercher et presser sur ok.
- ❹ Le systeme affiche un message d'erreur pour indique que c'est interdit de valider le choix vide.

☆ *Diagramme de Séquence :*

□ DESCRIPTION DU DIAGRAMME :

Le diagramme de séquence ci-dessus illustre le scénario de recherche des mots dans notre application. Au début, l'usager de l'application commence par la création du projet contrôle automatique d'une image. Ensuite, il demande l'outil de recherche des mots. Le système affiche la fenêtre de recherche pour que l'utilisateur introduit le mot a recherché. Après l'envoi du requête de recherche, le système procède à analyser le contenu du fichier ALTO par l'algorithme de recherche des mots.

A la fin de l'opération de recherche, notre système affiche les résultats de recherche dans l'onglet des résultats de recherche et si il y'a des mots trouvés, les boites englobant des mots trouvés sur l'image de la page traitée. L'usager peut également utiliser cet outil de recherche sur n'importe quel type de projet de notre application.

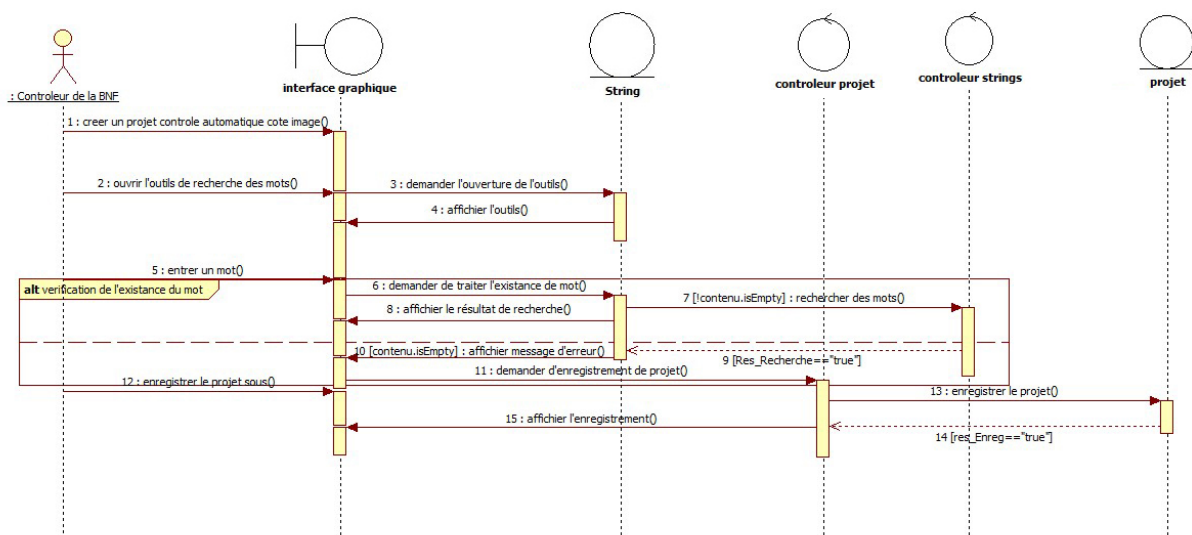


FIGURE 3.20 – Diagramme de séquence scénario8

③ Projet comparaison entre deux OCR cote répertoire d'images ‘

La deuxième fonctionnalité principale de notre application est la comparaison entre deux résultats de l'OCR. Cette opération de comparaison peut se faire entre

deux résultat de conversion d'une image ou entre deux résultats de conversion d'un document numérique (deux répertoires d'images). L'utilisateur de notre application commence par sélectionner le répertoire des images du document et les deux répertoires des fichiers ALTO produits par deux OCR différents (OCR1 et OCR2). Ensuite, l'interface contrôle la conformité de chaque nom de l'image avec le nom du fichier ALTO1 et le nom de fichier ALTO2.

Si tous les images du répertoire du document numérique ont deux fichiers ALTO dans les deux répertoires, Notre système affiche l'image avec les composants des deux fichiers ALTO (les paragraphes, les lignes, les mots et les illustrations). Les étapes d'ouverture de projet de comparaison entre deux OCR sont les suivants :

- ❶ L'usager de l'application crée un projet de comparaison entre deux répertoires ALTO
- ❷ Le système demande à l'utilisateur de choisir la répertoire d'images a traitée.
- ❸ L'utilisateur choisit la répertoire d'images.
- ❹ Le système prend en compte la répertoire d'images fournie par l'utilisateur.
- ❺ Le système demande à l'utilisateur la première répertoire des fichiers ALTO.
- ❻ L'utilisateur choisit la répertoire des fichiers ALTO.
- ❼ Le système vérifie la conformité du chaque nom de fichier ALTO avec le nom de chaque image.
- ❽ Le système demande à l'utilisateur de choisir deuxième répertoire des fichiers ALTO.
- ❾ L'utilisateur choisit la répertoire des fichiers ALTO.
- ❿ Le système vérifie la conformité du nom de fichier ALTO avec le nom de l'image et affiche la répertoire d'images et les éléments des deux Répertoires ALTO.

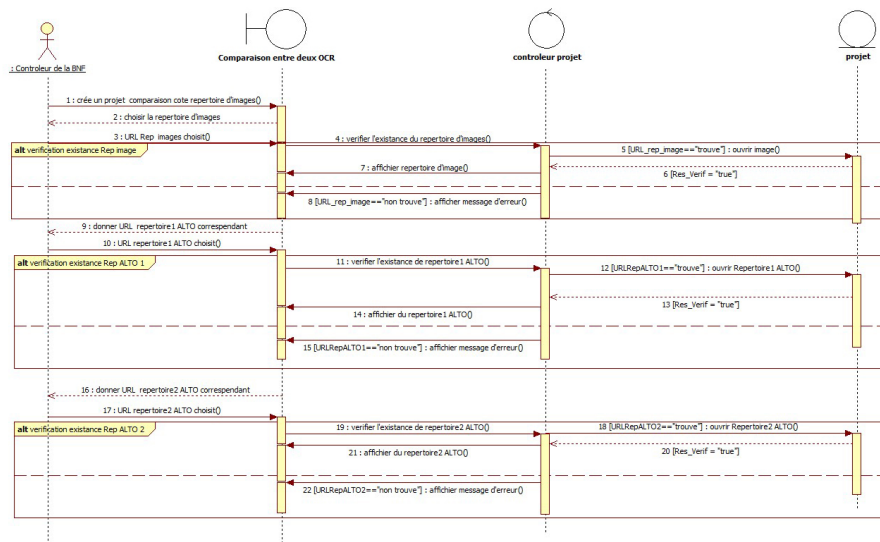


FIGURE 3.21 – Diagramme de séquence (création projet Comparaison entre deux OCR côté répertoire d'images)

❖ Scénario 9

L'utilisateur de l'application peut comparer les coordonnées des éléments de deux fichiers ALTO différents.

☆ Diagramme de cas d'utilisation

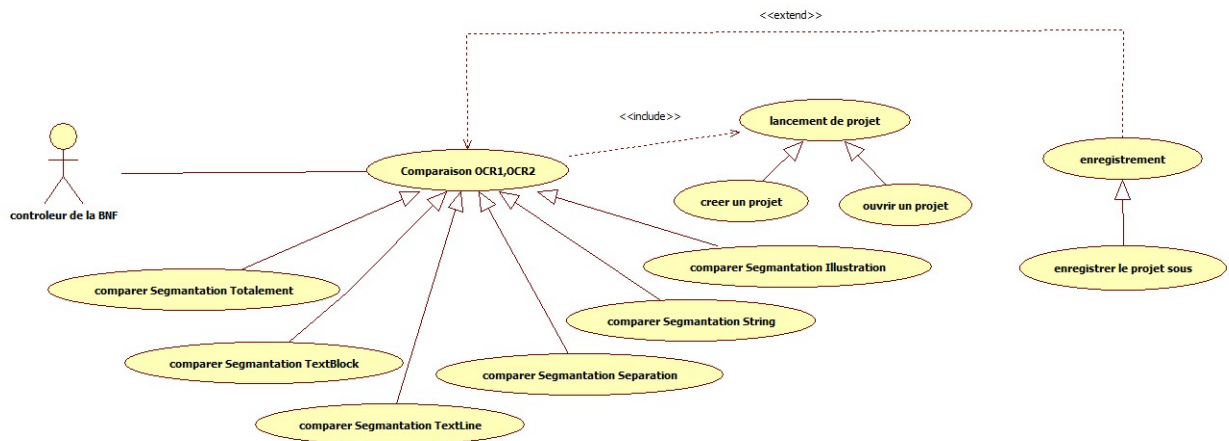


FIGURE 3.22 – Diagramme de cas d'utilisation scénario9

□ DESCRIPTION DU SCÉNARIO :

- **Nom du cas d'utilisation :** Comparaison entre deux OCR.
- **Acteur Principal :** Les contrôleurs de la Bibliothèque nationale de France.
- **Objectif :**
 - Comparer entre deux OCR.
- **Pré-condition :** L'utilisateur de l'application doit accéder aux outils de comparaison entre deux Ocr.
- **Post-condition :** Résultats de comparaison sont affichés.

□ SCÉNARIO DE TEST NOMINAL :

Après la création du projet, le contrôleur peut comparer les éléments de deux fichiers ALTOs. L'outil de comparaison se compose de plusieurs parties :

La partie de comparaison Total : l'utilisateur a la possibilité de comparer tous les éléments des deux fichiers ALTOs. C'est-à-dire, notre système permet de lancer une comparaison entre les paragraphes, les lignes, les mots, les espaces et les illustrations de deux fichiers ALTOs différents et afficher les résultats de comparaison sur l'interface graphique de notre application.

- **La partie de comparaison TextBlock :** l'utilisateur a la possibilité de comparer seulement les paragraphes des deux fichiers ATLO.

- **La partie de comparaison TextLine :** l'utilisateur a la possibilité de comparer seulement les lignes des deux fichiers ATLO.

- **La partie de comparaison String** : l'utilisateur a la possibilité de comparer seulement les mots des deux fichiers ATLO.
- **La partie de comparaison SP** : l'utilisateur a la possibilité de comparer seulement les espaces des deux fichiers ATLO.
- **La partie de comparaison Illustration** : l'utilisateur a la possibilité de comparer seulement les illustration des deux fichiers ATLO.

Les étapes de l'opération de comparaison sont :

L'utilisateur clique sur l'outil de comparaison entre les deux OCR.

Le système affiche tous les types de comparaison dans un menu des outils de comparaison :

- ❶ comparer Totalement
- ❷ comparer TextBlock
- ❸ comparer TextLine
- ❹ comparer String
- ❺ comparer SP
- ❻ comparer Illustration

L'utilisateur a la possibilité de choisir chaque type des éléments a comparés pour réaliser la comparaison entre ces éléments. Le système affiche les résultats de comparaison dans la fenêtre des résultats de comparaison et sur l'image si nous avons deux éléments différents.

☆ *Diagramme de Séquence :*

❑ DESCRIPTION DU DIAGRAMME :

Le diagramme de séquence ci-dessus illustre le scénario de comparaison entre deux OCR. Au début l'usager de l'application commence par crée un projet de comparaison entre deux OCR. Puis L'utilisateur choisit l'outil de comparaison entre deux OCR afin d'afficher le menu des outils de comparaison.

Les résultats de l'opération de comparaison sont affichés sur la fenêtre des résultats de comparaison et sur l'image du document. L'utilisateur peut également demander l'enregistrement des résultats de comparaison des deux fichiers ALTO. Dans ce cas, le système enregistre dans le fichier du projet les résultats de comparaison.

FIGURE 3.23 – Diagramme de séquence scénario9

4.3 Diagramme d'états-transitions

☆ *Definition :*

Les diagrammes d'états-transitions permettent de décrire les changements d'états d'un objet ou d'un composant, en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.

☆ *Gestion des diagrammes*

La couche métier de notre application se décompose en plusieurs modules. Chaque module se passe techniquement par un ensemble des états qui ont des rôles précis. L'objet **TextBlock** regroupe les mêmes états effectuer a chaque classes métiers en façon general.

□ Diagramme d'états-transitions du l'objet TextBlock

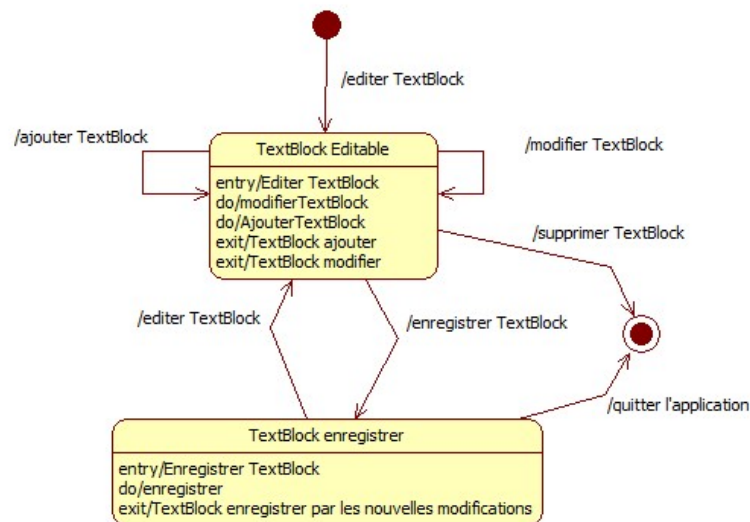


FIGURE 3.24 – Diagramme d'état-transitions de l'objet TextBlock

✱ **Explication de diagramme**

Le diagramme de la figure 3.20 présente le comportement simplifié du TextBlock, qui répond aux stimuli de deux boutons placés dans les menus de notre application. TextBlock peut-être dans deux états : Enregister , Editable. Lorsqu'il est editable, il est représentée par deux tâches dans le menu de correction de la segmentation. A l'etat normal, il peut être modifier ou ajouter. Lorsqu'il est sauvegardée , il permet de sauvegarder les parametre de l'objet TextBlock dans le fichier ALTO. L'état initial noté par un cercle plein, designe le point d'entrée du diagramme ou la création du l'objet. L'état final, désigné par un point dans un cercle correspond à la fin de vie de l'instance et à sa destruction.

✱ **Explication des états**

➤ **Etat Initial**

Cette état indique que la creation de l'objet TextBlock.

- **Etat Final** Cette état indique que l'objet TextBlock se termine ou se detruit.
- **Etat Intermédiaire (TextBlock Editable)** Cette etat indique que l'objet TextBlock se passer par un ensemble des mise a jour comme la modification et l'ajout de la boite englobant.
- **Etat Intermédiaire (TextBlock Enregistrer)** Cette etat indique que l'objet TextBlock s'enregistrer dans le fichier ALTO.

4.4 Diagramme de classe général de l'application

☆ *Definition :*

Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre classes. L'intérêt du diagramme de classe est de modéliser les entités du système d'information. Le diagramme de classe permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine.

Ces informations sont structurées, c'est-à-dire qu'elles ont regroupées dans des classes. Le diagramme met en évidence d'éventuelles relations entre ces classes. Le diagramme de classes comporte 6 concepts :

- **Classe**
- **Attribut**
- **Identifiant**
- **Relation**
- **Operation**
- **Généralisation / Spécialisation**

☆ *Explication du diagramme :*

Dans notre application nous avons appliqué une architecture Modèle/Vue/Contrôleur (MVC) pour organiser les classes de notre programme . Cette organisation consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface. Dans notre application, nous avons développé des classes modèle pour référencer les éléments du fichier ALTO (Page, PrintSpace, TextBloc...), des classes contrôle pour gérer les interactions entre les objets modèle et les objets interface.

Nous allons monter et expliquer dans les parties suivants les classes développés dans chaque type de classe.

□ Les classes

✿ **Modèle**

- Objet ALTO.
- Objet Projet.
- Objet PrintSpace.
- Objet TopMargin.
- Objet Page.
- Objet TextBlock.
- Objet ComposedBlock.
- Objet TextLine.
- Objet String.
- Objet Illustration.
- Objet SP.

❁ Contrôleur

- Contrôleur String.
- Contrôleur Segmentation.
- Contrôleur Comparaison.

❁ Vue

- interface graphique.
 - Interface Contrôle Automatique.
 - Interface Comparaison entre deux OCR.
- ❑ Les Attributs

✱ Modèle**➤ Objet ALTO**

Attributs	Type	Description
VecteurTextBlock	vector<TextBlock>	c'est le vecteur des objets TextBlock.
VecteurTextLine	vector<TextLine>	c'est le vecteur des objets TextLine.
VecteurString	vector<String>	c'est le vecteur des objets Strings.
VecteurSP	vector<SP>	c'est le vecteur des objets SP.
VecteurIllus	vector<Illustration>	c'est le vecteur des objets Illustrations.

TABLE 3.1 – Tableau des Attributs de l'objet ALTO

➤ Objet Projet

Attributs	Type	Description
Nom-ALTO1	String	c'est le nom du fichier ALTO1.
Nom-ALTO2	String	c'est le nom du fichier ALTO2.
Nom-Image	String	c'est le nom du Image.
Nom-Rep-ALTO1	String	c'est le nom de la première répertoire des fichiers ALTO.
Nom-Rep-ALTO2	String	c'est le nom de la deuxième répertoire des fichiers ALTO.
Nom-Rep-Image	String	c'est le nom de répertoire des images.

TABLE 3.2 – Tableau des Attributs de l'objet Projet

► Objet Page

Attributs	Type	Description
ID	String	c'est l'identifiant de la page du fichier ALTO.
HEIGHT	Integer	c'est la hauteur de la page.
WIDTH	Integer	c'est la largeur de la page.
PHYSICAL-IMG-NR	Integer	Indique le numéro de page du fichier ALTO.

TABLE 3.3 – Tableau des Attributs de l'objet Page

► Objet PrintSpace

Attributs	Type	Description
ID	String	c'est l'identifiant de PrintSpace.
HEIGHT	Integer	c'est la hauteur de PrintSpace.
WIDTH	Integer	c'est la largeur de PrintSpace.
HPOS	Integer	l'abscisse du coin supérieur gauche du bloc PrintSpace.
VPOS	Integer	l'ordonnée du coin supérieur gauche du bloc PrintSpace.

TABLE 3.4 – Tableau des Attributs de l'objet PrintSpace

► Objet TopMargin

Attributs	Type	Description
ID	String	c'est l'identifiant de TopMargin.
HEIGHT	Integer	c'est la hauteur de TopMargin.
WIDTH	Integer	c'est la largeur de TopMargin.
HPOS	Integer	l'abscisse du coin supérieur gauche du bloc TopMargin.
VPOS	Integer	l'ordonnée du coin supérieur gauche du bloc TopMargin.

TABLE 3.5 – Tableau des Attributs de l'objet TopMargin

► Objet ComposedBlock

Attributs	Type	Description
ID	String	c'est l'identifiant de composedBlock.
STYLEREFS	String	Référence aux styles de texte.
HEIGHT	Integer	c'est la hauteur de composedBlock.
WIDTH	Integer	c'est la largeur de composedBlock.
HPOS	Integer	l'abscisse du coin supérieur gauche du bloc composedBlock.
VPOS	Integer	l'ordonnée du coin supérieur gauche du bloc composedBlock.

TABLE 3.6 – Tableau des Attributs de l'objet composedBlock

► Objet TextBlock

Attributs	Type	Description
ID	String	c'est l'identifiant de TextBlock.
STYLEREFS	String	Référence aux styles de paragraphe
HEIGHT	Integer	c'est la hauteur de TextBlock.
WIDTH	Integer	c'est la largeur de TextBlock.
HPOS	Integer	l'abscisse du coin supérieur gauche du bloc TextBlock
VPOS	Integer	l'ordonnée du coin supérieur gauche du bloc TextBlock.

TABLE 3.7 – Tableau des Attributs de l'objet TextBlock

► Objet TextLine

Attributs	Type	Description
ID	String	c'est l'identifiant de TextLine.
STYLEREFS	String	Référence aux styles de TextLine
HEIGHT	Integer	c'est la hauteur de TextLine.
WIDTH	Integer	c'est la largeur de TextLine.
HPOS	Integer	l'abscisse du coin supérieur gauche du bloc TextLine.
VPOS	Integer	l'ordonnée du coin supérieur gauche du bloc TextLine.

TABLE 3.8 – Tableau des Attributs de l'objet TextLine

► Objet String

Attributs	Type	Description
ID	String	c'est l'identifiant d'objet String.
STYLEREFS	String	Référence aux styles d'objet String.
CONTENT	String	c'est le mot détecté par l'OCR.
HEIGHT	Integer	c'est la hauteur d'objet String.
WIDTH	Integer	c'est la largeur d'objet String.
HPOS	Integer	l'abscisse du coin supérieur gauche du bloc String.
VPOS	Integer	l'ordonnée du coin supérieur gauche du bloc String.

TABLE 3.9 – Tableau des Attributs de l'objet String

► Objet SP

Attributs	Type	Description
ID	String	c'est l'identifiant d'objet SP.
WIDTH	Integer	c'est la largeur d'objet SP.
HPOS	Integer	l'abscisse du coin supérieur gauche du bloc SP.
VPOS	Integer	l'ordonnée du coin supérieur gauche du bloc SP.

TABLE 3.10 – Tableau des Attributs de l'objet SP

► Objet Illustration

Attributs	Type	Description
ID	String	c'est l'identifiant d'objet Illustration.
HEIGHT	Integer	c'est la hauteur d'objet Illustration.
WIDTH	Integer	c'est la largeur d'objet Illustration.
HPOS	Integer	l'abscisse du coin supérieur gauche du bloc Illustration.
VPOS	Integer	l'ordonnée du coin supérieur gauche du bloc Illustration.

TABLE 3.11 – Tableau des Attributs de l'objet Illustration

□ Les Opérations

✱ Modèle

► Objet TextBlock

Opérations	Description
getID()	Cette méthode permet de retourner l'ID de l'objet TextBlock.
getSTYLEREFS()	Cette méthode permet de retourner un string qui indique la Référence aux styles de paragraphe.
getPOSH()	Cette méthode permet de retourner la position de TextBlock sur l'abscisse X.
getPOSV()	Cette méthode permet de retourner la position de TextBlock sur l'ordonnée Y.
getHEIGHT()	Cette méthode permet de retourner la hauteur de boîte englobant TextBlock.
getWidth	Cette méthode permet de retourner la largeur de boîte englobant TextBlock.
setPOSH(int POSH)	Cette méthode permet de remplacer l'ancien abscisse de coin supérieur gauche de TextBlock par une nouvelle valeur au paramètre de la fonction .
getPOSV(int POSV)	Cette méthode permet de remplacer l'ancien ordonné de coin supérieur gauche de TextBlock par une nouvelle valeur située au paramètre de la fonction .
setHEIGHT(int height)	Cette méthode permet de remplacer l'ancien hauteur de la boîte englobant TextBlock par une nouvelle valeur située au paramètre de la fonction
setWidth (int width)	Cette méthode permet de remplacer l'ancienne largeur de la boîte englobant TextBlock par une nouvelle valeur située au paramètre de la fonction.
AjouterSegTextLine (TextLine Tl)	Cette méthode permet d'ajouter des phrases sous un paragraphe sélectionné au cours de correction de la segmentation.
SupprimerSegTextLine (int numTl)	Cette méthode permet de supprimer des phrases mal segmentées par l'OCR.
ModifierSegTextLine (int numTl, TextLine Tl)	Cette méthode permet de modifier les phrases dans l'opération de correction de segmentation.

TABLE 3.12 – Tableau des Opérations de l'objet TextBlock

► **Objet TextLine**

Opérations	Description
getID()	Cette méthode permet de retourner l'ID de l'objet TextLine.
getSTYLEREFS()	Cette méthode permet de retourner un string qui indique la Référence aux styles de paragraphe.
getPOSH()	Cette méthode permet de retourner la position de l'objet TextLine sur l'abscisse X.
getPOSV()	Cette méthode permet de retourner la position de l'objet TextLine sur l'ordonnée Y.
getHeight()	Cette méthode permet de retourner la hauteur de l'objet TextLine.
getWidth	Cette méthode permet de retourner la largeur de l'objet TextLine.
setPOSH(int POSH)	cette méthode permet de remplacer l'abscisse de TextLine par une nouvelle valeur.
getPOSV(int POSV)	Cette méthode permet de remplacer l'ancien ordonné de coin supérieur gauche de TextLine par une nouvelle valeur
setHEIGHT(int height)	Cette méthode permet de remplacer l'ancien hauteur de la boîte englobant TextBlock par une nouvelle valeur.
setWidth (int width)	Cette méthode permet de remplacer l'ancienne largeur de la boîte englobant TextLine par une nouvelle valeur.
ajouterSegString()	Cette méthode permet d'ajouter des mots à chaque phrase au cours de correction de la segmentation.
SupprimerSegString()	Cette méthode permet de supprimer la segmentation des strings d'une phrase mal segmentée par l'OCR au cours de correction de la segmentation.
ModifierSegString()	Cette méthode permet de modifier les boîtes englobant les mots d'une phrase sélectionnée en utilisant l'outil de correction de la segmentation.
AjouterReconnaissanceString (int numS , String St)	Cette méthode permet d'ajouter les mots non reconnus par l'OCR dans la phrase en utilisant l'outil de correction des strings.

ModifierReconnaissanceString(String St)	Cette méthode permet de modifier les mots de la phrase encours d'utilisation qui sont mal reconnus par l'OCR.
SupprimerReconnaissanceString(int numS)	Cette méthode permet de supprimer les mots dans une phrase.

TABLE 3.13 – Tableau des Opérations de l'objet TextLine

► Objet String

Opérations	Description
getID()	Cette méthode permet de retourner l'ID de l'objet String.
getSTYLEREFS()	Cette méthode permet de retourner le style des mots.
getPOSH()	Cette méthode permet de retourner l'abscisse de String.
getPOSV()	Cette méthode permet de retourner l'ordonne de String.
getHEIGHT()	Cette méthode permet de retourner la hauteur de l'objet String.
getWidth	Cette méthode permet de retourner la largeur de l'objet String.
setPOSH(int POSH)	Cette méthode permet de remplacer l'ancienne abscisse de String par une nouvelle valeur.
getPOSV(int POSV)	cette méthode permet de remplacer l'ancien ordonne de String par une nouvelle valeur.
setHEIGHT(int height)	Cette méthode permet de remplacer l'ancien hauteur de l'objet String par une nouvelle valeur.
setWidth (int width)	Cette méthode permet de remplacer l'ancienne largeur de l'objet String par une nouvelle valeur.

TABLE 3.14 – Tableau des Opérations de l'objet String

► **SP (Séparation entre les mots)**

Opérations	Description
getID()	Cette méthode permet de retourner l'ID de l'objet SP.
getPOSH()	Cette méthode permet de retourner l'abscisse de SP.
getPOSV()	Cette méthode permet de retourner l'ordonnée de SP .
getWidth	Cette méthode permet de retourner la largeur de l'objet SP.
setPOSH(int POSH)	Cette méthode permet de remplacer l'ancienne abscisse de SP par une nouvelle valeur au paramètre de la fonction .
getPOSV(int POSV)	Cette méthode permet de remplacer l'ancien ordonnée de SP par une nouvelle valeur au paramètre de la fonction .
setWidth (int width)	Cette méthode permet de remplacer l'ancienne largeur de l'objet SP par une nouvelle valeur au paramètre de la fonction.

TABLE 3.15 – Tableau des Opérations de l'objet SP

► **Objet Illustration**

Opérations	Description
getID()	Cette méthode permet de retourner l'ID de l'objet Illustration.
getPOSH()	Cette méthode permet de retourner l'abscisse de Illustration.
getPOSV()	Cette méthode permet de retourner l'ordonnée de Illustration.
getHeight()	Cette méthode permet de retourner la hauteur de l'objet Illustration.
getWidth	Cette méthode permet de retourner la largeur de l'objet Illustration.

setPOSH(int POSH)	Cette méthode permet de remplacer l'ancienne abscisse de Illustration par une nouvelle valeur au paramètre de la fonction .
getPOSV(int POSV)	Cette méthode permet de remplacer l'ancien ordonné de Illustration par une nouvelle valeur au paramètre de la fonction .
setHEIGHT(int height)	Cette méthode permet de remplacer l'ancienne hauteur de l'objet Illustration par une nouvelle valeur au paramètre de la fonction .
setWidth (int width)	Cette méthode permet de remplacer l'ancienne largeur de l'objet Illustration par une nouvelle valeur au paramètre de la fonction.

TABLE 3.16 – Tableau des Opérations de l'objet Illustration

► **Objet ALTO**

Opérations	Description
getVecteurTextBlock()	Cette méthode permet de retourner le vecteur des objets TextBlock.
getVecteurIllustration()	Cette méthode permet de retourner le vecteur des objets Illustration.
ajouterObjTextBlock(TextBlock tb)	Cette méthode permet d'ajouter des objets.
supprimerObjTextBlock(int numTb)	Cette méthode permet de supprimer des objets TextBlock.
modifierTextBlock(TextBlock tb , int numTb)	Cette méthode permet de modifier des objets TextBlock.
ajouterObjIllustration(Illustration illus)	Cette méthode permet d'ajouter des objets Illustration.
supprimerObjIllustration (int numIllus)	Cette méthode permet de supprimer des objets Illustration.
modifierIllustration (Illustration tb , int numIllus)	Cette méthode permet de modifier des objets Illustration.
Enregistrer-sur-le-fichier()	Cette méthode permet d'enregistrer sur le fichier.

TABLE 3.17 – Tableau des Opérations de l'objet ALTO

► **Objet Projet**

Opérations	Description
EnregistrerProjetsous()	Cette méthode permet d'enregistrer le projet.

TABLE 3.18 – Tableau des Opérations de l'objet Projet

✱ **Controleur**

► **Contrôleur String**

Opérations	Description
controleaffString()	Cette méthode ordonne l'interface principale à afficher tous les mots de chaque phrase dans l'ALTO.
ajouterReconnaissanceString(int POSH, int POSV, int WIDTH, int HEIGHT)	Cette méthode ordonne le classe métier ALTO à ajouter des mots qui ne sont pas reconnu par l'OCR. Cette méthode s'effectue au cours de correction des strings.
supprimerReconnaissanceString()	Cette méthode ordonne la suppression des mots qui sont mal détecter par l'OCR. Cette méthode s'effectue au cours de correction des strings.
modifierReconnaissanceString (String content)	Cette méthode ordonne la modification des mots d'une phrase. Cette méthode s'effectue au cours de correction des strings.

TABLE 3.19 – Tableau des Opérations de Contrôleur String

► **Contrôleur Segmentation**

Opérations	Description
controleAffInitialSegTextBlock()	Cette méthode ordonne le contrôle de la tache d'affichage de Segmentation TextBlock.
controleAffInitialSegTextLine()	Cette méthode ordonne le contrôle de la tache d'affichage de Segmentation TextLine.
controleAffInitialSegString()	Cette méthode ordonne le contrôle de la tache d'affichage de segmentation String.

Opérations	Description
controleAffInitialSegSP()	Cette méthode ordonne le contrôle de la tâche d'affichage de segmentation de SP.
controleAffInitialSegIllustration()	Cette méthode ordonne le contrôle de la tâche d'affichage de segmentation de Illustration.
controleAjouterSegTextBlock (int POSH, int POSV, int WIDTH, int HEIGHT)	Cette méthode ordonne le contrôle de la méthode d'ajout de TextBlock cote correction de la segmentation.
controleSuppSegTextBlock ()	Cette méthode ordonne le contrôle de la méthode de suppression du TextBlock cote correction de la segmentation.
controleModSegTextBlock (int NPOSH, int NPOSV, int NWIDTH, int NHEIGHT)	Cette méthode ordonne le contrôle de la modification de la segmentation du TextBlock.
controleAjouterSegTextLine (int POSH, int POSV, int WIDTH, int HEIGHT)	Cette méthode ordonne le contrôle de la méthode d'ajout de TextLine cote correction de la segmentation.
controleSuppSegTextLine ()	Cette méthode ordonne le contrôle de la méthode de suppression du TextLine cote correction de la segmentation.
controleModSegTextLine (int NPOSH, int NPOSV, int NWIDTH, int NHEIGHT)	Cette méthode ordonne le contrôle de la modification de la segmentation du TextLine.
controleAjouterSegString (int POSH, int POSV, int WIDTH, int HEIGHT)	Cette méthode ordonne le contrôle de la méthode d'ajout de String cote correction de la segmentation.
controleSuppSegString ()	Cette méthode ordonne le contrôle de la méthode de suppression du String cote correction de la segmentation.
controleModSegString (int NPOSH, int NPOSV, int NWIDTH, int NHEIGHT)	Cette méthode ordonne le contrôle de la modification de la segmentation du String.
controleAjouterSegIllustration (int POSH, int POSV, int WIDTH, int HEIGHT)	Cette méthode ordonne le contrôle de la méthode d'ajout d'illustration cote correction de la segmentation.
controleSuppSegIllustration ()	Cette méthode ordonne le contrôle de la méthode de suppression du Illustration cote correction de la segmentation.
controleModSegIllustration(int NPOSH, int NPOSV, int NWIDTH, int NHEIGHT)	Cette méthode ordonne le contrôle de la modification de la segmentation du Illustration.

TABLE 3.20 – Tableau des Opérations de Contrôleur Segmentation

➤ **Contrôleur Comparaison**

Opérations	Description
comparerSegTotal(ALTO1 , ALTO2)	Cette méthode ordonne le contrôle de la méthode de comparaison entre les segmentations de toutes les objets (paragraphe , phrases ,mots, espace entre les mots , Illustration).
comparerSegTextBlock(ALTO1,ALTO2)	Cette méthode ordonne le contrôle de la méthode de comparaison entre les segmentations du objet TextBlock.
comparerSegTextLine(ALTO1,ALTO2)	Cette méthode ordonne le contrôle de la méthode de comparaison entre les segmentations du objet TextLine.
comparerSegString(ALTO1,ALTO2)	Cette méthode ordonne le contrôle de la méthode de comparaison entre les segmentations du String.
comparerSegSP (ALTO1,ALTO2)	Cette méthode ordonne le contrôle de la méthode de comparaison entre les segmentations du objet SP.
comparerIllustration(ALTO1,ALTO2)	Cette méthode ordonne le contrôle de la méthode de comparaison entre les segmentations du objet Illustration.

TABLE 3.21 – Tableau des Opérations de Contrôleur Comparaison

➤ **Contrôleur projet**

Opérations	Description
contrôle-enrg-sur-fichier ()	Cette méthode ordonne le contrôle de la méthode de la met a jour sur le fichier ALTO.
Contrôle-enreg-projet()	Cette méthode ordonne le contrôle de la méthode d'enregistrement de projet.

TABLE 3.22 – Tableau des Opérations de Contrôleur projet

✱ **Vue**

➤ **Interface graphique**

Opérations	Description
afficheControleSegInitialTextBlock()	Cette méthode permet d'afficher les boites englobant des objets TextBlock dans une but de vérification.
afficheControleSegInitialTextLine ()	Cette méthode permet d'afficher les boites englobant des objets TextLine dans une but de vérification.
afficheControleSegInitialString ()	Cette méthode permet d'afficher les boites englobant des objets String dans une but de vérification.
afficheControleSegInitialSP ()	Cette méthode permet d'afficher les boites englobant des objets SP dans une but de vérification.
afficheControleSegInitialIllustration ()	Cette méthode permet d'afficher les boites englobant des objets Illustrations dans une but de vérification.
AfficheRecherche()	Cette méthode permet d'afficher la résultats de recherche des mots dans le fichier ALTO.
AfficheContent()	Cette méthode permet d'afficher les mots du fichier ALTO pour vérifier les scripts des mots.
AfficheMiseajourTb ()	Cette méthode permet d'afficher les résultats de tous les mises à jour (modification, suppression, ajout) effectués sur l'objet TextBlock.
AfficheMiseajourTl()	Cette méthode permet d'afficher les résultats de tous les mises à jour (modification, suppression, ajout) effectués sur l'objet TextLine.
AfficheMiseajourSt()	cette méthode permet d'afficher les résultats de tous les mises à jour (la modification, la suppression, l'ajout) effectués sur l'objet String.
AfficheMiseajourIllus()	Cette méthode permet d'afficher les résultats des mises à jour (la modification, la suppression, l'ajout) effectués sur l'objet Illustration.
AfficheEnreg()	Cette methode permet d'afficher les résultats d'enregistrement de projet ou de fichier.

TABLE 3.23 – Tableau des Opérations de l'interface graphique

► Interface Contrôle Automatique

La classe **interface de contrôle automatique** hérite toutes les méthodes de classe **interface graphique principale**

► Interface Comparaison entre deux OCR

La classe **interface de Comparaison entre deux OCR** hérite toutes les méthodes de classe **interface graphique principale** et nous trouvons aussi d'autres méthodes spécifiques comme :

Opérations	Description
AfficheComparaisonSegTotal()	Cette méthode permet d'afficher le résultat de comparaison de tous les objets.
AfficheComparaisonSegTextBlock()	Cette méthode permet d'afficher les résultats de comparaison de la segmentation des paragraphes.
AfficheComparaisonSegTextLine()	Cette méthode permet d'afficher les résultats de comparaison de la segmentation des phrases.
AfficheComparaisonSegString()	Cette méthode permet d'afficher les résultats de comparaison de la segmentation des strings.
AfficheComparaisonSegSP()	Cette méthode permet d'afficher les résultats de comparaison de la segmentation des SP.
AfficheComparaisonSegIllus()	Cette méthode permet d'afficher les résultats de comparaison de la segmentation des illustrations.

TABLE 3.24 – Tableau des Opérations de l'interface Comparaison entre deux OCR

□ Les Relations

✱ Composition

Une classe composée est détruite lorsque sa classe mère disparaît. Dans notre cas, les classes de modèle sont liées avec une relation de composition. Donc, lors de la disparition d'une classe mère la classe composée est détruite. Par exemple, les TextLine (les phrases) sont inclus dans un TextBlock (un paragraphe). La suppression d'un paragraphe dans l'opération de contrôle des résultats de l'OCR entraîne la suppression de toutes les phrases qui appartiennent à ce paragraphe.

✱ Association Direct ou navigabilité

Les classes contrôleur exercent plusieurs opérations de contrôle sur les classes de la couche métier et sur les classes de la couche vue mais il ne contrôle pas au même temps la même classe. Le type de la relation (1.n) montre que le contrôleur peut exécuter plusieurs fois ces tâches sur les classes du modèle ou de vue. Cette relation est appelée **une relation de navigabilité**.

FIGURE 3.25 – Diagramme de classe général de l'application

5 Conclusion

Dans ce chapitre on a défini les méthodologies à utiliser et la démarche reproductible pour obtenir des résultats fiables. Le chapitre suivant met en évidence la pratique de la conception et les différents résultats du développement de l'application.

Chapitre 4

Réalisation

1 Introduction

Nous avons vu au cours des chapitres précédents les différents aspects conceptuels de notre application et les scénarios d'utilisation qui nous ont permis par la suite de définir les fonctionnalités et les classes des trois couches du modèle MVC.

Nous présentons dans ce chapitre les aspects techniques mis en œuvre pour l'accomplissement du projet à savoir l'environnement de travail dans lequel va être réalisée l'application, les choix techniques adoptés ainsi qu'une exposition de la phase d'implémentation et des tests.

2 Environnement de travail

Dans cette partie nous allons décrire l'ensemble des outils matériels et logiciels que nous avons utilisé pour développer notre application.

2.1 Environnement matériel

Etant donné que l'application développée dans le cadre de ce projet appartient au genre des applications client lourd. Notre application a été implémentée et testée dans un ordinateur portable qui a les caractéristiques suivantes :

- Equipé d'un processeur Intel® Core(TM) i7 CPU. 1.6 GHz
- Doté d'une mémoire de 4 GO.
- Muni d'un disque dur généreux 400 Go.

2.2 Environnement de logiciel

✱ Choix de framework multiplateforme. (Qt Creator 2009.03 SDK)

Notre choix , en ce qui concerne la bibliothèques multiplateforme , est d'adopter avec la langage de programmation C++ (Visual Studio C++). Nous avons utilisé cet framework pour les raisons suivantes :

- Qt est un framework multiplateforme. (Figure 4.1)

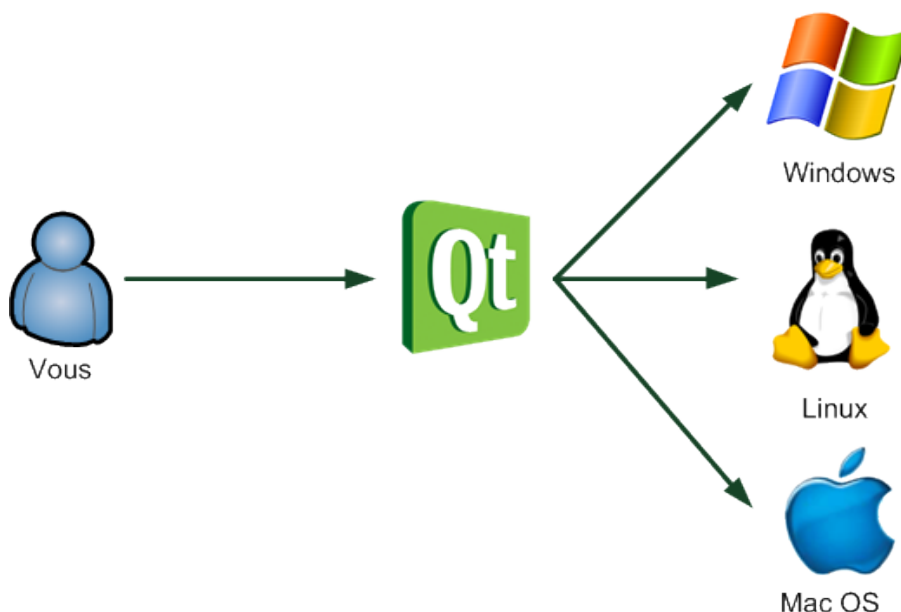


FIGURE 4.1 – Le fonctionnement de Qt

- Tout d’abord, il simplifie grandement la création d’une fenêtre. En effet, il faut beaucoup moins de lignes de code pour ouvrir une **simple** fenêtre.

- Ensuite, il uniformise le tout, elles forment un ensemble cohérent qui fait qu’il est facile de s’y retrouver. Les noms des fonctions et des classes sont choisis de manière logique de manière à nous aider autant que possible.

- Enfin, notre interface est une partie d’un projet de recherche dans lequel les algorithmes de contrôle et de vérification automatique des résultats de l’OCR sont programmés avec la bibliothèque Open CV. Afin de faciliter l’opération d’intégration de ces algorithmes, nous avons choisi QT pour développer notre application

- Qt est donc constituée d’un ensemble de bibliothèques, appelées **modules**. On peut y trouver entre autres ces fonctionnalités :

- **Module GUI** : c’est toute la partie création de fenêtres.
- **Module OpenGL** : Qt peut ouvrir une fenêtre contenant de la 3D gérée par OpenGL.
- **Module de dessin** : pour tous ceux qui voudraient dessiner dans leur fenêtre (en 2D).
- **Module réseau** : Qt fournit une batterie d’outils pour accéder au réseau.
- **Module SVG** : possibilité de créer des images et animations vectorielles, à la manière de Flash.
- **Module de script** : Qt supporte le Javascript (ou ECMAScript).
- **Module XML** : pour ceux qui connaissent le XML, c’est un moyen très pratique d’échanger des données avec des fichiers formés à l’aide de balises, un peu comme le XHTML.
- **Module SQL** : permet un accès aux bases de données (MySQL, Oracle, PostgreSQL...).

Dans notre travail, nous avons utilisé le module GUI, le module de dessin et le module XML.

3 Interfaces graphiques

Les contrôleurs de la BnF ont un niveau intellectuel respectable et des connaissances moyennes en informatique cela nous amène à leur concevoir des interfaces graphiques ergonomiques et faciles à utiliser. En effet, nous avons respecté toujours la règle de trois dans la conception des différentes interfaces de notre application.

3.1 Interface de contrôle Automatique

L'interface de contrôle automatique offre à les utilisateurs la possibilité de visualiser de contrôler une ou plusieurs images. En effet, elle contient des outils d'affichage des éléments du fichier ALTO sur l'images du document et des outils de contrôle et de vérification qui permet de modifier, supprimer , ajouter les éléments manquants ou incorrects et de rechercher des mots.

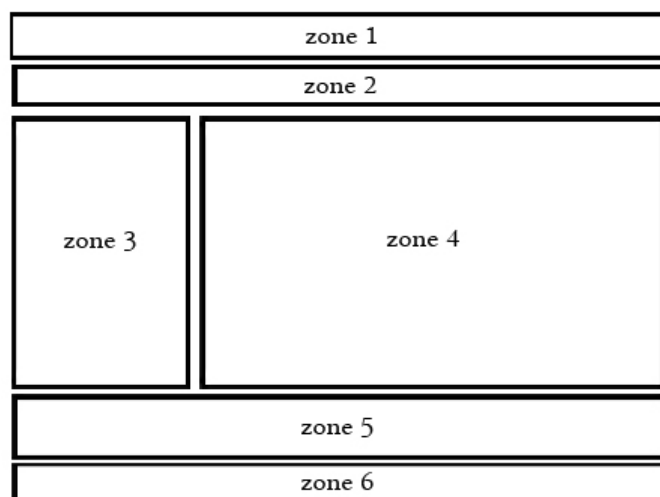


FIGURE 4.2 – Schéma de l'interface de contrôle automatique côté répertoire d'images

Comme le montre la figure 4.2, l'interface de contrôle automatique d'un document numérique est formée de 6 zones :

- **Zone 1** : Elle représente la barre de menu qui est visible en permanence et qui offre la possibilité de naviguer à tout moment entre les menus.

☆ *Menu Fichier* :

☞ **Action créer un projet** : Cette action donne la possibilité à l'utilisateur de choisir son type de projet (contrôle automatique ou comparaison entre deux fichiers ALTOs).

☞ **Action Imprimer** : Cette action donne la possibilité à l'utilisateur d'imprimer l'image en cours de traitement.

☞ **Action Réinitialiser** : Cette action représente la partie de restauration du logiciel.

☞ **Action Ouvrir un projet** : Cette action permet à l'utilisateur d'ouvrir des anciens projets.

☞ **Action Enregistrer** : Cette action donne la possibilité à l'utilisateur de sauvegarder ses traitements dans le répertoire courant du fichier ALTO.

☞ **Action Enregistrer sous** : Cette action permet de sélectionner le répertoire ou l'utilisateur veut enregistrer ses modifications.

Les figures 4.3 et 4.4 montrent un exemple d'un **projet contrôle automatique d'un répertoire d'images**

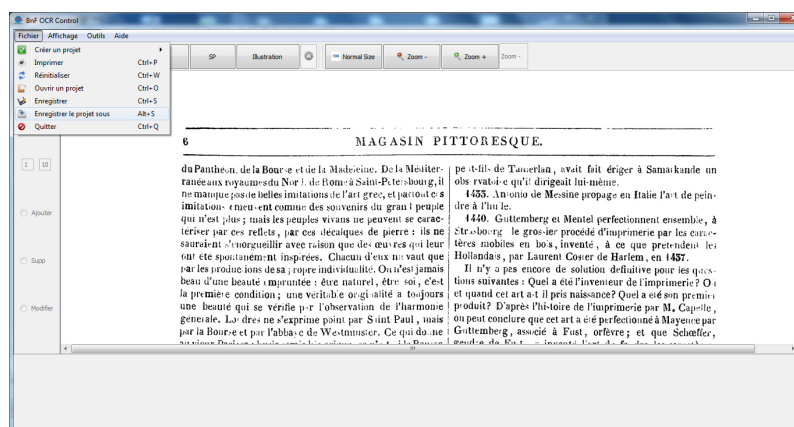


FIGURE 4.3 – Interface de enregistrer sous le projet

Les informations conservées dans le fichier du projet de contrôle automatique d'un document numérique sont présentées dans la figure ??.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Description_Enregistrement Logiciel="Bnf OCR Control version 1.0" type_traitement="Contrôle Automatique côté repertoire d'images">
  <Image_ALTO nom_image="Repertoire d'images" url_image="C:/Users/Achraf/Desktop/Bnf_OCR_Control/debug/">
    <Fichier_ALTO url_Fichier="C:/Users/Achraf/Desktop/Bnf_OCR_Control/debug/fwd (2)" nom_Fichier="Repertoire xml"/>
    <Fichier_Enregistrement nom_Enregistrement="desc" url_Enregistrement="C:/Users/Achraf/Desktop/desc.xml"/>
  </Image_ALTO>
</Description_Enregistrement>
```

FIGURE 4.4 – Enregistrement du projet

☞ **Action Quitter** : Cette action lance l'opération de fermeture de logiciel.

☆ *Menu Affichage* :

☞ **Action Zoom avant** : Cette action permet à l'utilisateur de **agrandir** l'affichage de l'image.

☞ **Action Zoom arrière** : Cette action permet à l'utilisateur de **réduire** l'affichage de l'image.

☞ **Action affichage normal** : Cette action affiche l'image avec sa taille normale.

☞ **Action font d'écran** : Cette action construit un affichage de l'image en font d'écran .

☆ *Menu Outils* :

✎ **Outils correction des strings :** Cette outils offre à l'utilisateur la possibilité de corriger les erreurs des mots reconnus par l'OCR.

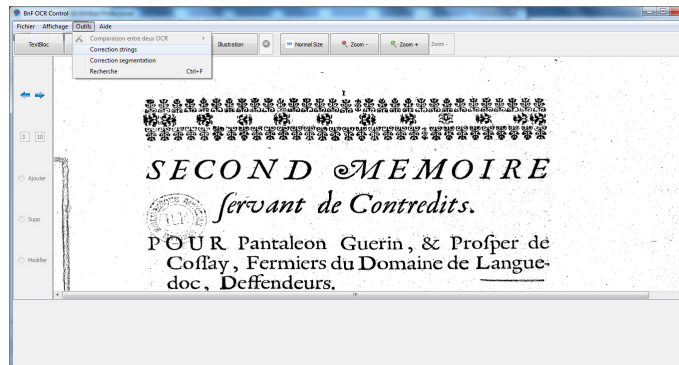


FIGURE 4.5 – Outils correction des strings

→ La figure 4.5 indique l'emplacement de outils de correction des mots.

★ **Méthode d'ajout des strings :**

❖ Avant l'ajout

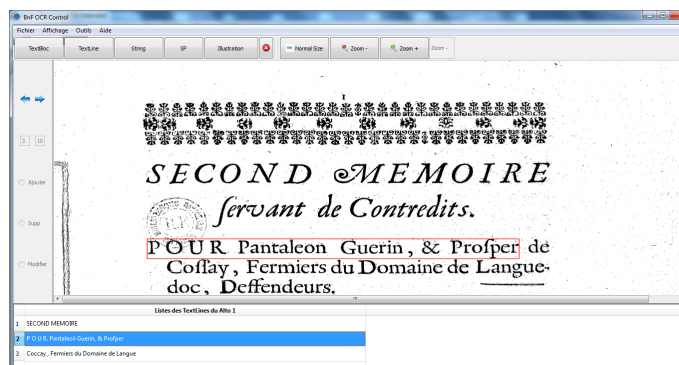


FIGURE 4.6 – Avant l'ajout du mot dans l'application

→ Selon la figure 4.6, l'utilisateur commence par sélectionner une TextLine pour ajouter les mot qui ne sont pas détectés par l'OCR. Dans l'exemple de la figure 4.6 nous avons ajouté le mot "de" dans la phrase traitée.

```
<TextLine WIDTH="1590" HEIGHT="98" ID="PAG_00000001_TL000005" HPOS="400" STYLEREFS="TXT_5" VPOS="1329">
  <String CONTENT="P" WIDTH="55" HEIGHT="78" ID="PAG_00000001_ST000006" HPOS="400" STYLEREFS="TXT_5" VPOS="1333"/>
  <String CONTENT="O" WIDTH="163" HEIGHT="77" ID="PAG_00000001_ST000007" HPOS="481" STYLEREFS="TXT_6" VPOS="1330"/>
  <String CONTENT="U" WIDTH="70" HEIGHT="69" ID="PAG_00000001_ST000008" HPOS="670" STYLEREFS="TXT_5" VPOS="1337"/>
  <String CONTENT="R." WIDTH="379" HEIGHT="70" ID="PAG_00000001_ST000009" HPOS="790" STYLEREFS="TXT_5" VPOS="1333"/>
  <String CONTENT="Pantaleoi" WIDTH="308" HEIGHT="82" ID="PAG_00000001_ST000010" HPOS="1228" STYLEREFS="TXT_5" VPOS="1337"/>
  <String CONTENT="Guerin," WIDTH="76" HEIGHT="73" ID="PAG_00000001_ST000011" HPOS="1576" STYLEREFS="TXT_5" VPOS="1332"/>
  <String CONTENT="s" WIDTH="295" HEIGHT="97" ID="PAG_00000001_ST000012" HPOS="1695" STYLEREFS="TXT_5" VPOS="1330"/>
  <String CONTENT="Profper" WIDTH="50" HEIGHT="50" ID="Ajouter" HPOS="1990" STYLEREFS="TXT_1" VPOS="1330"/>
  <SP WIDTH="28" ID="PAG_00000001_SP000002" HPOS="454" VPOS="1330"/>
  <SP WIDTH="28" ID="PAG_00000001_SP000003" HPOS="643" VPOS="1329"/>
  <SP WIDTH="52" ID="PAG_00000001_SP000004" HPOS="739" VPOS="1334"/>
  <SP WIDTH="61" ID="PAG_00000001_SP000005" HPOS="1168" VPOS="1332"/>
  <SP WIDTH="42" ID="PAG_00000001_SP000006" HPOS="1535" VPOS="1332"/>
  <SP WIDTH="45" ID="PAG_00000001_SP000007" HPOS="1651" VPOS="1331"/>
</TextLine>
```

FIGURE 4.7 – Avant l'ajout du mot dans le fichier ALTO

- La figure 4.7 montre les éléments de fichier ALTO avant l'ajout du mot.
- ❖ Après l'ajout

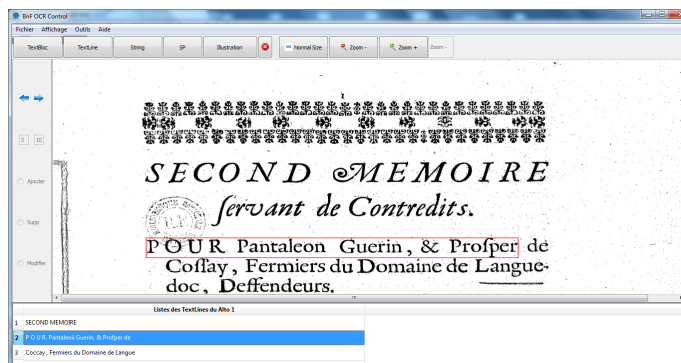


FIGURE 4.8 – Après l'ajout du mot dans l'application

- Cette figure 4.8 affiche la résultat de l'opération d'ajout du mot sur l'interface graphique de notre application.

```
<TextLine WIDTH="1590" HEIGHT="98" ID="PAG_00000001_TL000005" HPOS="400" STYLEREFS="TXT_5" VPOS="1329">
  <String CONTENT="P" WIDTH="55" HEIGHT="78" ID="PAG_00000001_ST000006" HPOS="400" STYLEREFS="TXT_5" VPOS="1333"/>
  <String CONTENT="O" WIDTH="163" HEIGHT="77" ID="PAG_00000001_ST000007" HPOS="481" STYLEREFS="TXT_6" VPOS="1330"/>
  <String CONTENT="U" WIDTH="70" HEIGHT="69" ID="PAG_00000001_ST000008" HPOS="670" STYLEREFS="TXT_5" VPOS="1337"/>
  <String CONTENT="R." WIDTH="379" HEIGHT="70" ID="PAG_00000001_ST000009" HPOS="790" STYLEREFS="TXT_5" VPOS="1333"/>
  <String CONTENT="Pantaleoi" WIDTH="308" HEIGHT="82" ID="PAG_00000001_ST000010" HPOS="1228" STYLEREFS="TXT_5" VPOS="1337"/>
  <String CONTENT="Guerin," WIDTH="76" HEIGHT="73" ID="PAG_00000001_ST000011" HPOS="1576" STYLEREFS="TXT_5" VPOS="1332"/>
  <String CONTENT="e" WIDTH="295" HEIGHT="97" ID="PAG_00000001_ST000012" HPOS="1695" STYLEREFS="TXT_5" VPOS="1330"/>
  <String CONTENT="Profper" WIDTH="50" HEIGHT="50" ID="Ajouter" HPOS="1990" STYLEREFS="TXT_1" VPOS="1330"/>
  <String CONTENT="de" WIDTH="50" HEIGHT="50" ID="Ajouter" HPOS="2040" STYLEREFS="TXT_1" VPOS="1330"/>
  <SP WIDTH="28" ID="PAG_00000001_SP000002" HPOS="454" VPOS="1330"/>
  <SP WIDTH="28" ID="PAG_00000001_SP000003" HPOS="643" VPOS="1329"/>
  <SP WIDTH="52" ID="PAG_00000001_SP000004" HPOS="739" VPOS="1334"/>
  <SP WIDTH="61" ID="PAG_00000001_SP000005" HPOS="1168" VPOS="1332"/>
  <SP WIDTH="42" ID="PAG_00000001_SP000006" HPOS="1535" VPOS="1332"/>
  <SP WIDTH="45" ID="PAG_00000001_SP000007" HPOS="1651" VPOS="1331"/>
</TextLine>
```

FIGURE 4.9 – Après l'ajout du mot dans le fichier ALTO

- La figure 4.9 montre la bonne prise en compte du mot ajouté par l'utilisateur dans le fichier ALTO.

★ Méthode de suppression des strings :

Dans les fichiers ALTO, nous pouvons trouver un ensemble des erreurs l'hors de l'opération de reconnaissance des caractères. Pour cela, nous avons présenté dans cette partie, l'interface qui sert à supprimer les mots incorrects.

❖ Avant Suppression

- La figure 4.10 montre les mots reconnus par l'OCR avant l'opération de suppression.

- Cette figure 4.11 montre les mots qui sont exister dans le fichier ALTO.

❖ Après Suppression

- La figure 4.12 montre le résultat visuel de l'opération de suppression des mots
- Cette figure 4.13 affiche la suppression des mots sélectionner côte fichier XML.

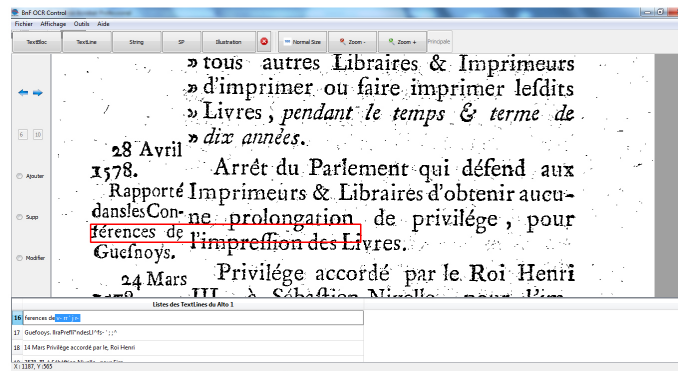


FIGURE 4.10 – Avant la suppression du mot à travers notre interface de notre application

```

▼<TextLine WIDTH="541" HEIGHT="36" ID="PAG_00000001_TL000016" HPOS="78" STYLEREFS="TXT_6" VPOS="880">
  <String CONTENT="ferences" WIDTH="131" HEIGHT="30" ID="PAG_00000001_ST000098" HPOS="78" STYLEREFS="TXT_8" VPOS="880"/>
  <String CONTENT="de" WIDTH="33" HEIGHT="28" ID="PAG_00000001_ST000099" HPOS="231" STYLEREFS="TXT_8" VPOS="881"/>
  <String CONTENT="v-" WIDTH="31" HEIGHT="20" ID="PAG_00000001_ST000100" HPOS="279" STYLEREFS="TXT_2" VPOS="895"/>
  <String CONTENT="rr" WIDTH="33" HEIGHT="20" ID="PAG_00000001_ST000101" HPOS="419" STYLEREFS="TXT_6" VPOS="896"/>
  <String CONTENT="j" WIDTH="7" HEIGHT="4" ID="PAG_00000001_ST000102" HPOS="470" STYLEREFS="TXT_6" VPOS="893"/>
  <String CONTENT="r-" WIDTH="16" HEIGHT="19" ID="PAG_00000001_ST000103" HPOS="517" STYLEREFS="TXT_6" VPOS="897"/>
  <String CONTENT="r-" WIDTH="35" HEIGHT="18" ID="PAG_00000001_ST000104" HPOS="584" STYLEREFS="TXT_2" VPOS="898"/>
  <SP WIDTH="24" ID="PAG_00000001_SP000083" HPOS="208" VPOS="881"/>
  <SP WIDTH="17" ID="PAG_00000001_SP000084" HPOS="263" VPOS="881"/>
  <SP WIDTH="111" ID="PAG_00000001_SP000085" HPOS="309" VPOS="895"/>
  <SP WIDTH="20" ID="PAG_00000001_SP000086" HPOS="451" VPOS="893"/>
  <SP WIDTH="42" ID="PAG_00000001_SP000087" HPOS="476" VPOS="893"/>
  <SP WIDTH="53" ID="PAG_00000001_SP000088" HPOS="532" VPOS="896"/>
</TextLine>

```

FIGURE 4.11 – Avant la suppression du mot dans le fichier ALTO

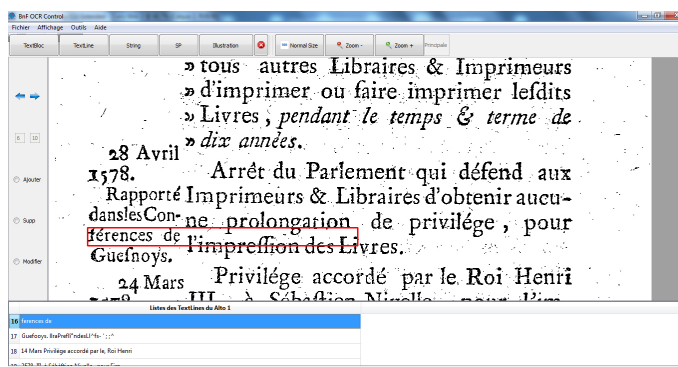


FIGURE 4.12 – Après la suppression du mot à travers l'interface graphique de notre application

```

▼<TextLine WIDTH="541" HEIGHT="36" ID="PAG_00000001_TL000016" HPOS="78" STYLEREFS="TXT_6" VPOS="880">
  <String CONTENT="ferences" WIDTH="131" HEIGHT="30" ID="PAG_00000001_ST000098" HPOS="78" STYLEREFS="TXT_8" VPOS="880"/>
  <String CONTENT="de" WIDTH="33" HEIGHT="28" ID="PAG_00000001_ST000099" HPOS="231" STYLEREFS="TXT_8" VPOS="881"/>
  <SP WIDTH="24" ID="PAG_00000001_SP000083" HPOS="208" VPOS="881"/>
  <SP WIDTH="17" ID="PAG_00000001_SP000084" HPOS="263" VPOS="881"/>
  <SP WIDTH="111" ID="PAG_00000001_SP000085" HPOS="309" VPOS="895"/>
  <SP WIDTH="20" ID="PAG_00000001_SP000086" HPOS="451" VPOS="893"/>
  <SP WIDTH="42" ID="PAG_00000001_SP000087" HPOS="476" VPOS="893"/>
  <SP WIDTH="53" ID="PAG_00000001_SP000088" HPOS="532" VPOS="896"/>
</TextLine>

```

FIGURE 4.13 – Après la suppression du mot dans le fichier ALTO

★ Méthode de modification des strings :

L'utilisateur sélectionne les lignes de l'éditeur pour modifier et corriger les mots incorrects. Chaque ligne sélectionnée dans l'éditeur doit être affichée sur l'image de la page pour faciliter l'opération de vérification et de modification.

✦ Avant Modification

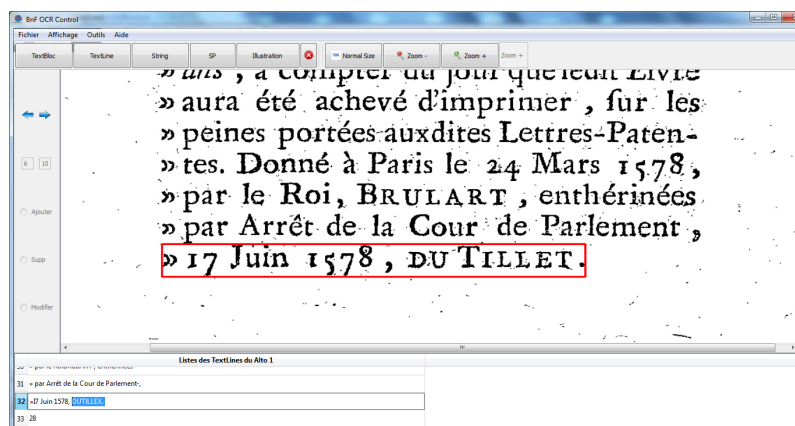


FIGURE 4.14 – Avant la modification du mot à travers l'interface graphique de notre application

→ La figure 4.14 montre le mot incorrect que le contrôleur de la BnF veut le modifier à travers l'interface graphique de notre application.

```
<TextLine WIDTH="606" HEIGHT="52" ID="PAG_00000001_TL000032" HPOS="276" STYLEREFS="TXT_3" VPOS="1653">
<String CONTENT="»17" WIDTH="80" HEIGHT="38" ID="PAG_00000001_ST000212" HPOS="276" STYLEREFS="TXT_3" VPOS="1667"/>
<String CONTENT="Juin" WIDTH="84" HEIGHT="41" ID="PAG_00000001_ST000213" HPOS="375" STYLEREFS="TXT_2" VPOS="1653"/>
<String CONTENT="1578," WIDTH="122" HEIGHT="50" ID="PAG_00000001_ST000214" HPOS="485" STYLEREFS="TXT_1" VPOS="1655"/>
<String CONTENT="DUTILLEX." WIDTH="253" HEIGHT="39" ID="PAG_00000001_ST000215" HPOS="629" STYLEREFS="TXT_3" VPOS="1656"/>
<SP WIDTH="21" ID="PAG_00000001_SP000181" HPOS="355" VPOS="1653"/>
<SP WIDTH="28" ID="PAG_00000001_SP000182" HPOS="458" VPOS="1654"/>
<SP WIDTH="24" ID="PAG_00000001_SP000183" HPOS="606" VPOS="1656"/>
</TextLine>
```

FIGURE 4.15 – Avant la modification du mot côté fichier XML

→ La figure 4.15 montre le contenu du fichier ALTO avant l'opération de modification du mot.

✦ Après Modification

→ La figure 4.16 montre les nouvelles modifications effectuées sur les mots de la ligne qui est encours de traitement.

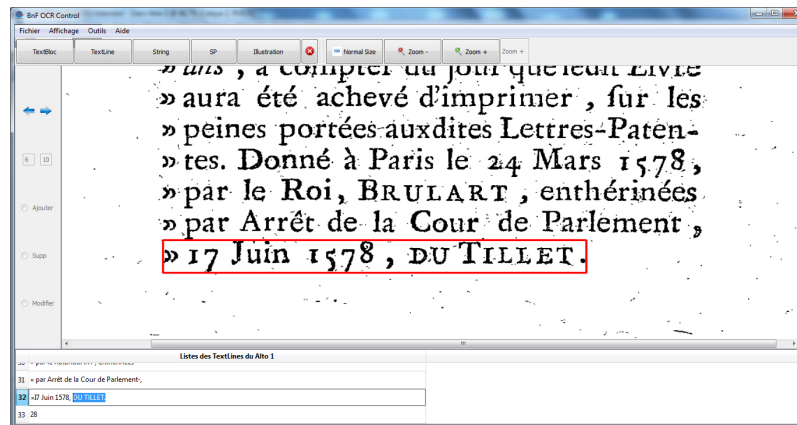


FIGURE 4.16 – Après la modification du mot à travers l'interface graphique de notre application

```
<TextLine WIDTH="606" HEIGHT="52" ID="PAG_00000001_TL000032" HPOS="276" STYLEREFS="TXT_3" VPOS="1653">
  <String CONTENT="»I7" WIDTH="80" HEIGHT="38" ID="PAG_00000001_ST000212" HPOS="276" STYLEREFS="TXT_3" VPOS="1667"/>
  <String CONTENT="Juin" WIDTH="84" HEIGHT="41" ID="PAG_00000001_ST000213" HPOS="375" STYLEREFS="TXT_2" VPOS="1653"/>
  <String CONTENT="1578," WIDTH="122" HEIGHT="50" ID="PAG_00000001_ST000214" HPOS="485" STYLEREFS="TXT_1" VPOS="1655"/>
  <String CONTENT="DU" WIDTH="253" HEIGHT="39" ID="PAG_00000001_ST000215" HPOS="629" STYLEREFS="TXT_3" VPOS="1656"/>
  <String CONTENT="TILLET." WIDTH="50" HEIGHT="50" ID="Ajouter" HPOS="882" STYLEREFS="TXT_1" VPOS="1656"/>
  <SP WIDTH="21" ID="PAG_00000001_SP000181" HPOS="355" VPOS="1653"/>
  <SP WIDTH="28" ID="PAG_00000001_SP000182" HPOS="458" VPOS="1654"/>
  <SP WIDTH="24" ID="PAG_00000001_SP000183" HPOS="606" VPOS="1656"/>
</TextLine>
```

FIGURE 4.17 – Le résultat de l'opération de modification des mots qui existent dans le fichier ALTO

→ La figure 4.17 montre les mots corrigés dans le fichier ALTO suite à l'opération de modifications des défauts de reconnaissance des mots.

Outils correction segmentation : L'interface de correction de la segmentation contient des outils adaptés pour corriger les boîtes englobantes de chaque élément du fichier ALTO. A travers les outils proposés dans notre application, les contrôleurs de la BnF peuvent corriger la segmentation incorrecte des paragraphes (**TextBlock**), des phrases (**TextLine**), des mots (**String**) et des illustrations (**Illustration**).

★ Opération d'ajout des boîtes englobant des éléments manqués :

Pour simuler l'opération d'ajout des blocs de type **String**, **TextBlock** et **TextLine**, nous avons commencé par l'ajout d'un paragraphe (**TextBlock**) et les phrases (**TextLine**) qui vont contenir les mots (**String**) manqués. Ensuite grâce à une opération simple de glissement de souris, l'utilisateur procède à l'ajout des mots.

❖ Opération d'ajout d'un paragraphe (**TextBlock**)

Dans cette partie nous allons présenter l'opération d'ajout des boîtes englobantes "**TextBlock**" sur l'interface graphique et les résultats de cette opération dans le fichier ALTO.

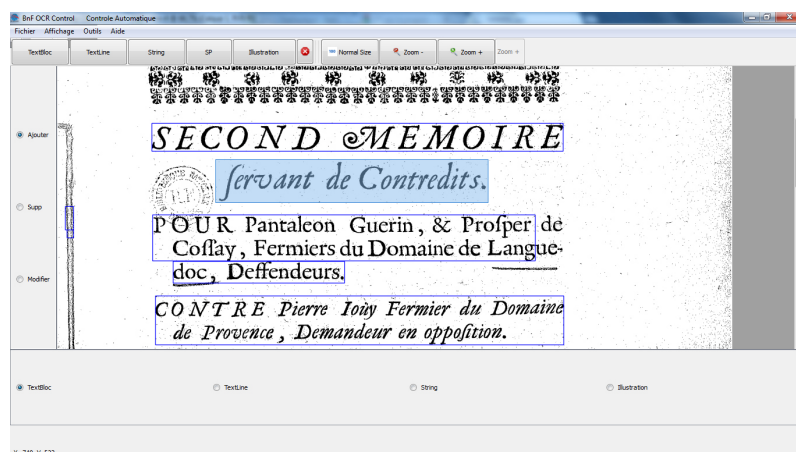


FIGURE 4.18 – L'opération d'ajout d'un TextBlock à travers notre interface graphique

```

▼<TextBlock WIDTH="55" HEIGHT="25" ID="PAG_00000001_TB000014" HPOS="2129" STYLEREFS="TXT_13" VPOS="4314">
  ▼<TextLine WIDTH="55" HEIGHT="25" ID="PAG_00000001_TL000046" HPOS="2129" STYLEREFS="TXT_12" VPOS="4314">
    <String CONTENT="i" WIDTH="10" HEIGHT="25" ID="PAG_00000001_ST000444" HPOS="2129" STYLEREFS="TXT_12" VPOS="4314"/>
    <String CONTENT="1" WIDTH="8" HEIGHT="17" ID="PAG_00000001_ST000445" HPOS="2147" STYLEREFS="TXT_9" VPOS="4321"/>
    <String CONTENT="n" WIDTH="17" HEIGHT="14" ID="PAG_00000001_ST000446" HPOS="2167" STYLEREFS="TXT_12" VPOS="4323"/>
    <SP WIDTH="10" ID="PAG_00000001_SP000399" HPOS="2138" VPOS="4314"/>
    <SP WIDTH="14" ID="PAG_00000001_SP000400" HPOS="2154" VPOS="4321"/>
  </TextLine>
</TextBlock>
<TextBlock WIDTH="1142" HEIGHT="185" ID="PAG_00000001_TB15" HPOS="664" STYLEREFS="TXT_15" VPOS="1096"/>

```

FIGURE 4.19 – Le résultat de l'opération d'ajout des paragraphes dans le fichier ALTO

❖ Opération d'ajout d'une ligne (TextLine)

Ici nous allons montrer l'opération d'ajout d'une nouvelle ligne **TextLine**. Comme nous avons mentionné au début de cette partie, nous pouvons ajouter une ligne que à l'intérieur d'un paragraphe.

❖ Opération d'ajout d'un mot (String)

Les boites englobantes des mots ne peut être ajoutées que à l'intérieur d'une boite englobante d'une ligne. L'opération d'insertion d'une boite "mot" passe par deux phases principales :

- phase de traçage de la boite englobante,
- phase de détermination des caractères du mot.

Nous allons montrer dans les parties suivantes une simulation de l'opération d'ajout d'un mot dans la phrase ajoutée dans la section précédente.

❶ Phase de traçage de la boite englobante

→ La figure 4.22 simule l'opération de marquage de la boite englobante d'un mot oublié par l'OCR.

❷ Phase de détermination des caractères du mot

→ La figure 4.23 montre la fenêtre qui permet d'annoter le contenu du mot après l'opération de traçage de la boite englobante.

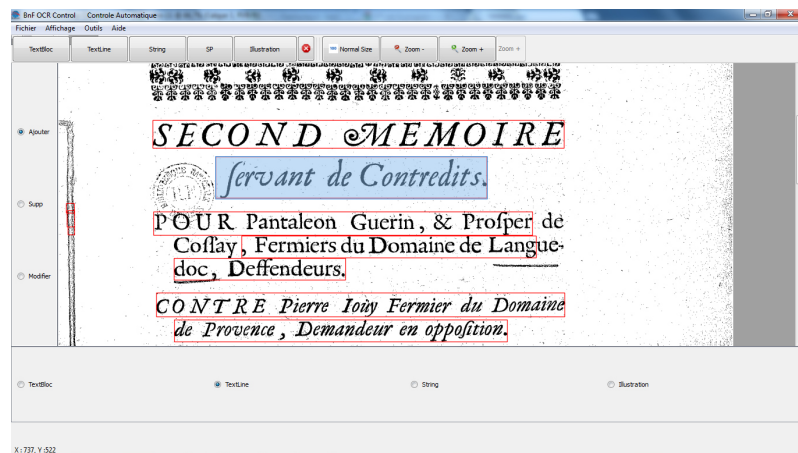


FIGURE 4.20 – Opération d'ajout d'une ligne (TextLine) à travers l'interface graphique de notre application

```

▼<TextBlock WIDTH="55" HEIGHT="25" ID="PAG_00000001_TB000014" HPOS="2129" STYLEREFS="TXT_13" VPOS="4314">
  ▼<TextLine WIDTH="55" HEIGHT="25" ID="PAG_00000001_TL000046" HPOS="2129" STYLEREFS="TXT_12" VPOS="4314">
    <String CONTENT="1" WIDTH="8" HEIGHT="17" ID="PAG_00000001_ST000445" HPOS="2147" STYLEREFS="TXT_9" VPOS="4321"/>
    <String CONTENT="n" WIDTH="17" HEIGHT="14" ID="PAG_00000001_ST000446" HPOS="2167" STYLEREFS="TXT_12" VPOS="4323"/>
    <SP WIDTH="10" ID="PAG_00000001_SP000399" HPOS="2138" VPOS="4314"/>
    <SP WIDTH="14" ID="PAG_00000001_SP000400" HPOS="2154" VPOS="4321"/>
  </TextLine>
</TextBlock>
▼<TextBlock WIDTH="1142" HEIGHT="185" ID="PAG_00000001_TB15" HPOS="664" STYLEREFS="TXT_15" VPOS="1096">
  <TextLine WIDTH="1137" HEIGHT="178" ID="Ajouter" HPOS="659" STYLEREFS="TXT_2" VPOS="1096"/>
</TextBlock>

```

FIGURE 4.21 – Le résultat d'ajout des lignes dans le fichier ALTO

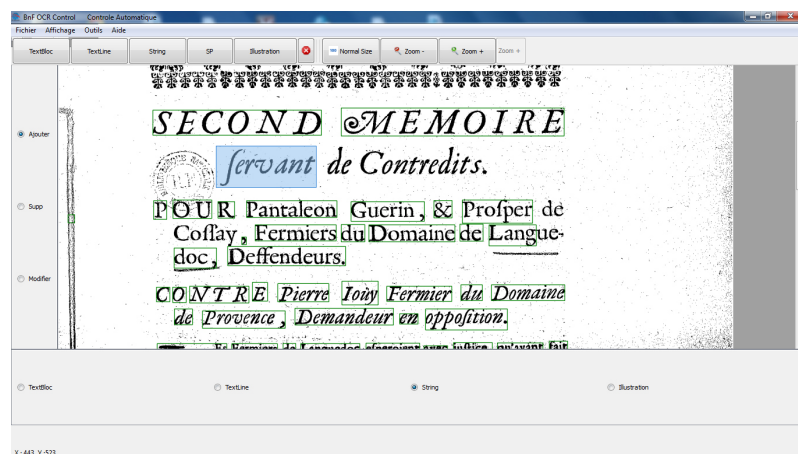


FIGURE 4.22 – Ajouter string (phase de traçage de la boîte englobante d'un mot)

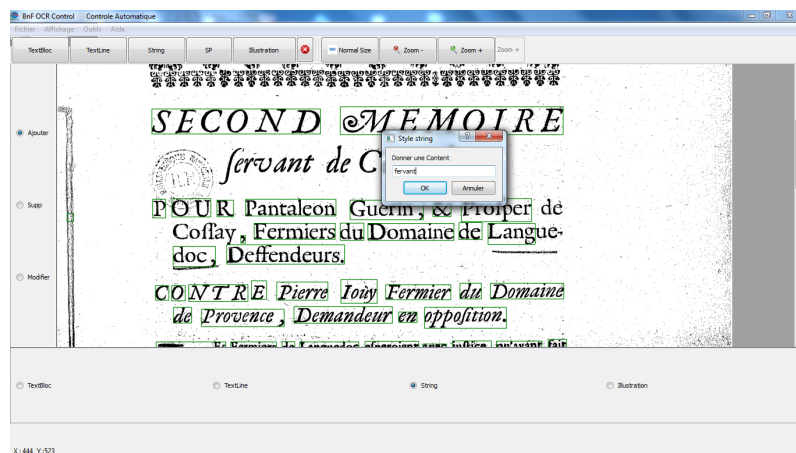


FIGURE 4.23 – Ajouter string (phase d'annotation de la contenu d'un mot)

```

▼<TextBlock WIDTH="55" HEIGHT="25" ID="PAG_00000001_TB000014" HPOS="2129" STYLEREFS="TXT_13" VPOS="4314">
  ▼<TextLine WIDTH="55" HEIGHT="25" ID="PAG_00000001_TL000046" HPOS="2129" STYLEREFS="TXT_12" VPOS="4314">
    <String CONTENT="i" WIDTH="10" HEIGHT="25" ID="PAG_00000001_ST000444" HPOS="2129" STYLEREFS="TXT_12" VPOS="4314"/>
    <String CONTENT="1" WIDTH="8" HEIGHT="17" ID="PAG_00000001_ST000445" HPOS="2147" STYLEREFS="TXT_9" VPOS="4321"/>
    <String CONTENT="n" WIDTH="17" HEIGHT="14" ID="PAG_00000001_ST000446" HPOS="2167" STYLEREFS="TXT_12" VPOS="4323"/>
    <SP WIDTH="10" ID="PAG_00000001_SP000399" HPOS="2138" VPOS="4314"/>
    <SP WIDTH="14" ID="PAG_00000001_SP000400" HPOS="2154" VPOS="4321"/>
  </TextLine>
</TextBlock>
▼<TextBlock WIDTH="1142" HEIGHT="185" ID="PAG_00000001_TB15" HPOS="664" STYLEREFS="TXT_15" VPOS="1096">
  ▼<TextLine WIDTH="1137" HEIGHT="178" ID="Ajouter" HPOS="659" STYLEREFS="TXT_2" VPOS="1096">
    <String CONTENT="servant" WIDTH="407" HEIGHT="170" ID="AjouterString" HPOS="676" STYLEREFS="TXT_2" VPOS="1105"/>
  </TextLine>
</TextBlock>

```

FIGURE 4.24 – Résultats de l'opération d'ajout des mots dans le fichier ALTO.

★ Opération de suppression des boîtes englobantes :

Parmi les outils de correction de segmentation qu'ils sont mis à la disposition des contrôleurs de la BnF, nous trouvons l'outil de suppression du bruit de détection. Dans la simulation de cette opération nous allons montrer un scénario de suppression des paragraphes (**TextBlock**) supplémentaires.

```

▼<Page WIDTH="2832" HEIGHT="4443" pagePHYSICAL_IMG_NR="" ID="PAG_00000001">
▼<PrintSpace WIDTH="2446" HEIGHT="3457" ID="PAG_00000001 PrintSpace" HPOS="29" VPOS="945">
▼<TextBlock WIDTH="35" HEIGHT="98" ID="PAG_00000001_TB000001" HPOS="34" STYLEREFS="TXT_13" VPOS="1294">
  ▼<TextLine WIDTH="34" HEIGHT="40" ID="PAG_00000001_TL000001" HPOS="34" STYLEREFS="TXT_1" VPOS="1294">
    <String CONTENT="P" WIDTH="34" HEIGHT="40" ID="PAG_00000001_ST000001" HPOS="34" STYLEREFS="TXT_1" VPOS="1294"/>
  </TextLine>
  ▼<TextLine WIDTH="32" HEIGHT="68" ID="PAG_00000001_TL000002" HPOS="37" STYLEREFS="TXT_2" VPOS="1324">
    <String CONTENT="P" WIDTH="32" HEIGHT="68" ID="PAG_00000001_ST000002" HPOS="37" STYLEREFS="TXT_2" VPOS="1324"/>
  </TextLine>
</TextBlock>
▼<TextBlock WIDTH="28" HEIGHT="33" ID="PAG_00000001_TB000002" HPOS="41" STYLEREFS="TXT_13" VPOS="1392">
  ▼<TextLine WIDTH="28" HEIGHT="33" ID="PAG_00000001_TL000003" HPOS="41" STYLEREFS="TXT_3" VPOS="1392">
    <String CONTENT="fi" WIDTH="28" HEIGHT="33" ID="PAG_00000001_ST000003" HPOS="41" STYLEREFS="TXT_3" VPOS="1392"/>
  </TextLine>
</TextBlock>

```

FIGURE 4.25 – L'élément incorrect "TextBlock" dans le fichier ALTO

→ La figure 4.25 montre un fichier ALTO qui contient un élément TextBlock incorrect.

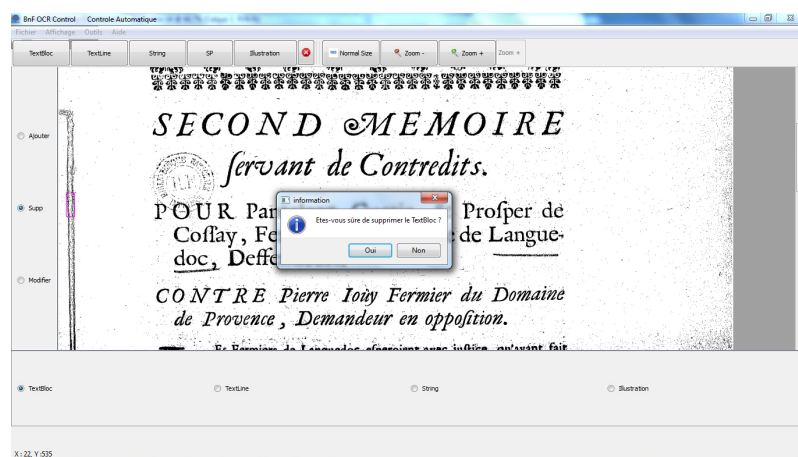


FIGURE 4.26 – Suppression de la boîte englobante TextBlock

→ La figure 4.26 simule l'opération de suppression de la boîte englobante **TextBlock** à l'aide de l'interface graphique de notre application.

```

▼<Page WIDTH="2832" HEIGHT="4443" pagePHYSICAL_IMG_NR="" ID="PAG_00000001">
▼<PrintSpace WIDTH="2446" HEIGHT="3457" ID="PAG_00000001 PrintSpace" HPOS="29" VPOS="945">
▼<TextBlock WIDTH="28" HEIGHT="33" ID="PAG_00000001_TB000002" HPOS="41" STYLEREFS="TXT_13" VPOS="1392">
  ▼<TextLine WIDTH="28" HEIGHT="33" ID="PAG_00000001_TL000003" HPOS="41" STYLEREFS="TXT_3" VPOS="1392">
    <String CONTENT="fi" WIDTH="28" HEIGHT="33" ID="PAG_00000001_ST000003" HPOS="41" STYLEREFS="TXT_3" VPOS="1392"/>
  </TextLine>
</TextBlock>
▼<TextBlock WIDTH="1725" HEIGHT="117" ID="PAG_00000001_TB000003" HPOS="397" STYLEREFS="TXT_13" VPOS="945">
  ▼<TextLine WIDTH="1725" HEIGHT="117" ID="PAG_00000001_TL000004" HPOS="397" STYLEREFS="TXT_4" VPOS="945">
    <String CONTENT="SECOND" WIDTH="707" HEIGHT="105" ID="PAG_00000001_ST000004" HPOS="397" STYLEREFS="TXT_4" VPOS="954"/>
    <String CONTENT="MEMOIRE" WIDTH="935" HEIGHT="117" ID="PAG_00000001_ST000005" HPOS="1187" STYLEREFS="TXT_4" VPOS="945"/>
    <SP WIDTH="85" ID="PAG_00000001_SP000001" HPOS="1103" VPOS="947"/>
  </TextLine>
</TextBlock>

```

FIGURE 4.27 – La position du TextBlock après suppression

→ La figure 4.27 montre la résultat de suppression de la boite englobante incorrecte dans le fichier ALTO.

★ Opération de modification des boites englobante :

Dans cette partie, nous allons parler de l'opération de modification des coordonnées des boites englobantes de l'illustration **Illustration**.

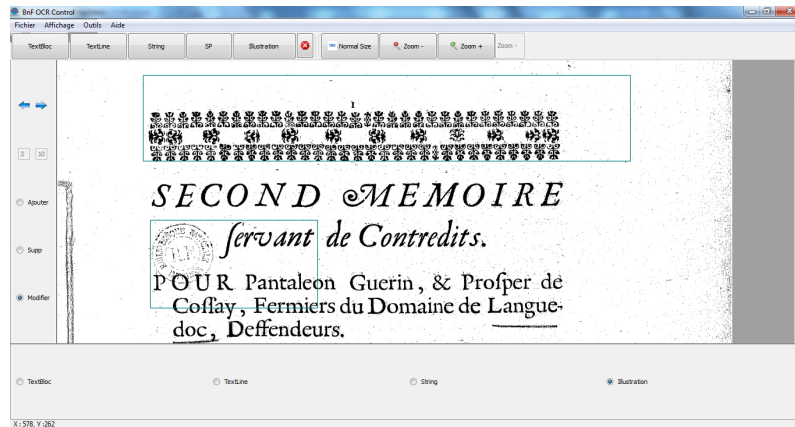


FIGURE 4.28 – l’affichage des boites englobantes Illustration avant l’opération de modification

→ La figure 4.28 montre les boites englobantes incorrectes qui englobent les Illustrations avant l’opération de modification de ses coordonnées.

```
<Illustration WIDTH="2044" HEIGHT="359" ID="PAG_00000001_illus1" HPOS="361" VPOS="503"/>
<Illustration WIDTH="701" HEIGHT="368" ID="PAG_00000001_illus2" HPOS="391" VPOS="1110"/>
</PrintSpace>
</Page>
```

FIGURE 4.29 – Le contenu du fichier ALTO avant l’opération de modification des coordonnées des boites englobantes

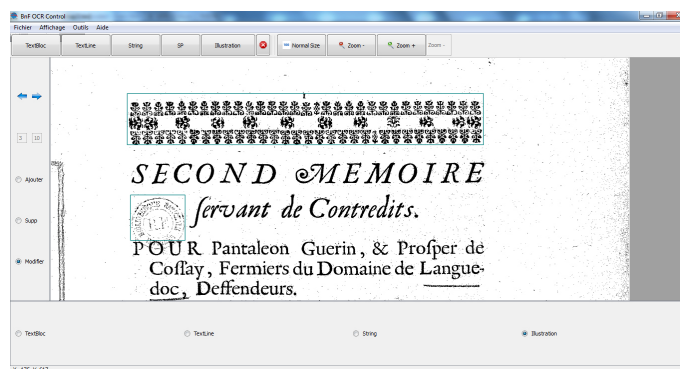


FIGURE 4.30 – Les boites englobantes "Illustrations" après l’opération de modification de ses coordonnées

→ La figure 4.30 montre les nouvelles coordonnées des boites qui englobent les Illustrations dans le fichier ALTO

```

<Illustration WIDTH="1742" HEIGHT="250" ID="PAG_00000001_illus1" HPOS="375" VPOS="615"/>
<Illustration WIDTH="267" HEIGHT="222" ID="PAG_00000001_illus2" HPOS="391" VPOS="1110"/>
</PrintSpace>
</Page>

```

FIGURE 4.31 – Le contenu du fichier ALTO après l'opération de modification des boîtes englobant

🔍 **Outils Recherche :** Le contrôleur de la BnF peut également rechercher des mots dans le contenu du fichier ALTO à travers un outil de recherche adapté. Les étapes de recherche des mots sont les suivantes :

L'utilisateur sélectionne l'outil de recherche à travers le menu de l'application ou à travers le raccourci clavier "Ctrl+F".



FIGURE 4.32 – Etape d'ouverture de l'outils

Après l'affichage de la fenêtre de recherche, l'utilisateur tape le mot à rechercher.



FIGURE 4.33 – Etape de détermination du mot

Les résultats de l'algorithme de recherche des mots sont affichés dans la figure 4.34

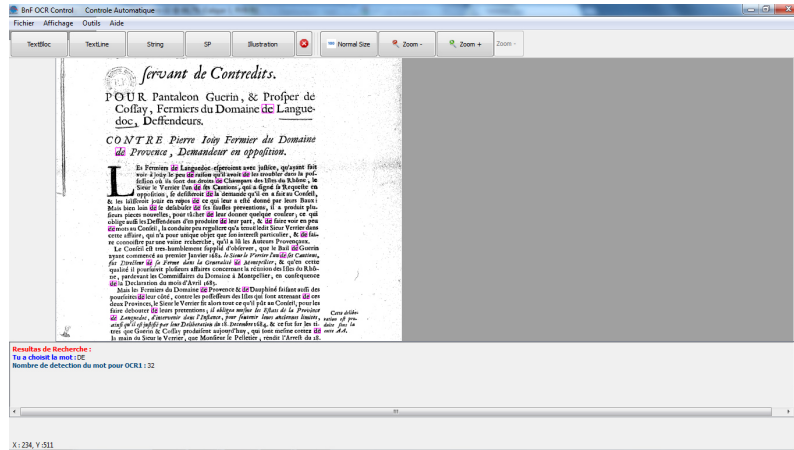


FIGURE 4.34 – Etape de détection du mot

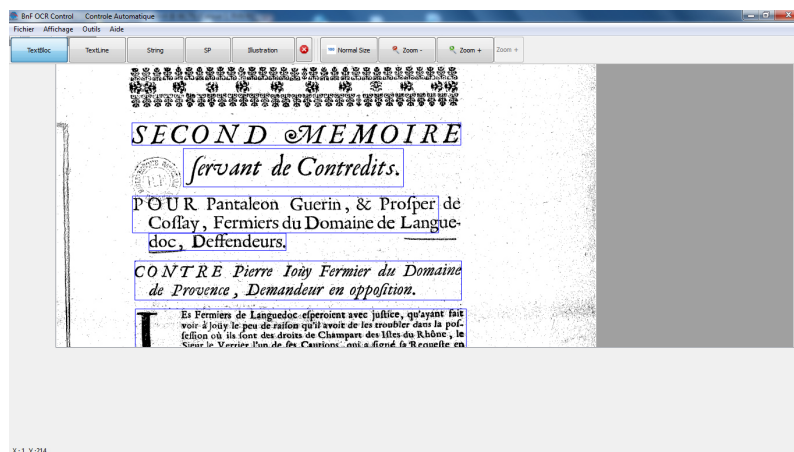


FIGURE 4.35 – Affichage TextBlock

3.2 Interface de comparaison entre deux OCR

L'interface de comparaison entre deux fichier ALTOs offrent à l'utilisateur la possibilité de comparer entre deux résultats d'OCR appliqué sur une seule image ou sur un répertoire d'images à l'aide des outils de comparaison avancée.

Ces outils aident le contrôleur à déterminer les différences qui existent entre les deux résultats de segmentation des paragraphes **Comparaison TextBlock** , des lignes **Comparaison TextLine** , des mots **Comparaison String** et des illustrations **Comparaison Illustration**.

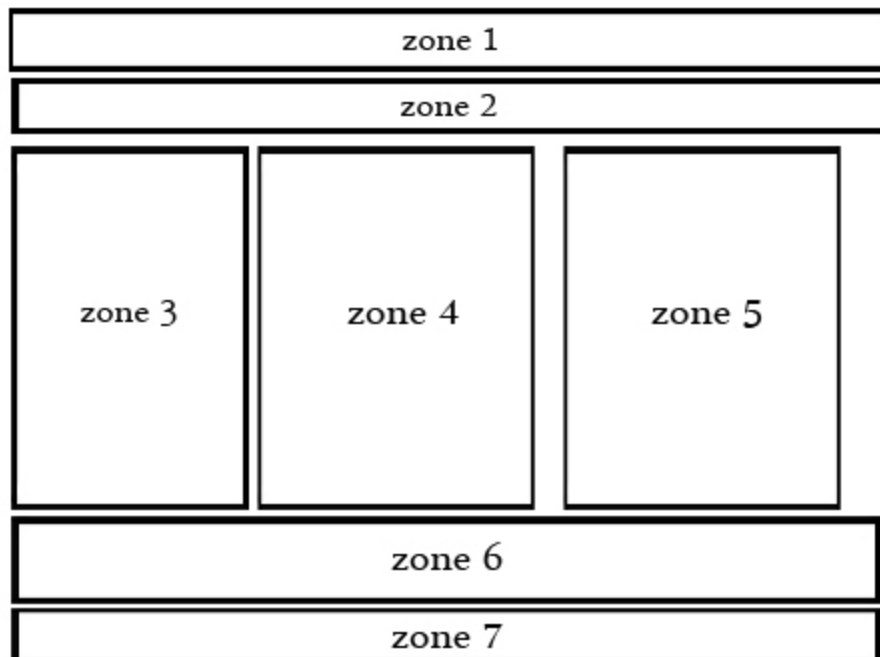


FIGURE 4.36 – Schéma de l'interface de comparaison entre deux OCR

Conformément à l'interface contrôle automatique, les zones 1,2,3,6 et 7 de l'interface comparaison entre deux fichiers ALTOs ont les même fonctionnalités. Par contre, **la zone 4** dans l'interface de comparaison est réservé pour afficher les éléments du premier fichier ALTO. La zone 5 est réservé pour afficher les éléments du deuxième fichier ALTO et les résultats visuels de comparaison (figure 4.36) .

Nous avons utilisé deux fichiers ALTO pour simuler l'opération de comparaison entre ces deux résultats d'OCR. Nous avons comparé ici les coordonnées des boites englobantes **TextBlock**

La figure 4.38 représente le résultat de l'opération de comparaison entre deux fichiers ALTO. Les rectangles pointillés en rouge représentent les éléments du premier fichier ALTO qui ne sont différents des éléments du deuxième fichier ALTO. Le nombre des éléments différents est affiché dans la zone 6. Grâce à la représentation simultanée des éléments des deux fichiers ALTO et le résultat quantitative de l'opération de comparaison le contrôleur de la BnF est capable de juger la qualité de segmentation de chaque fichier ALTO.

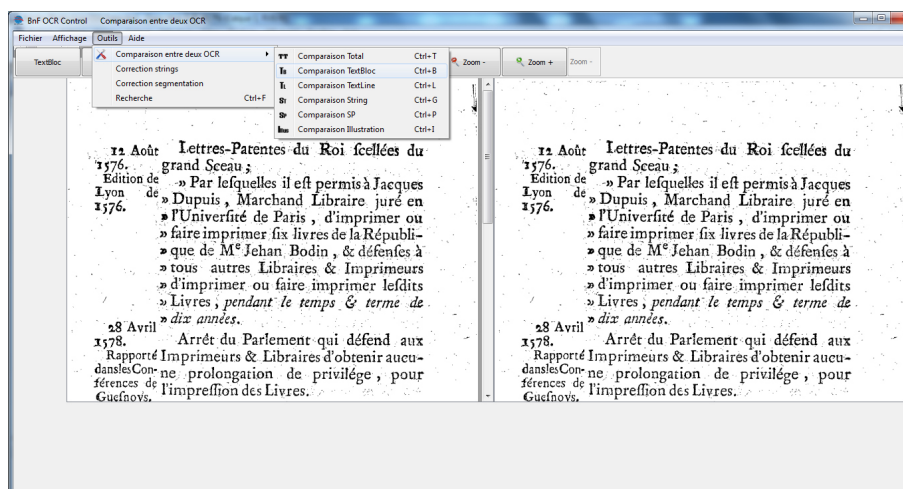


FIGURE 4.37 – Comparaison entre les TextBlocks de deux fichier ALTO (Etape 1)

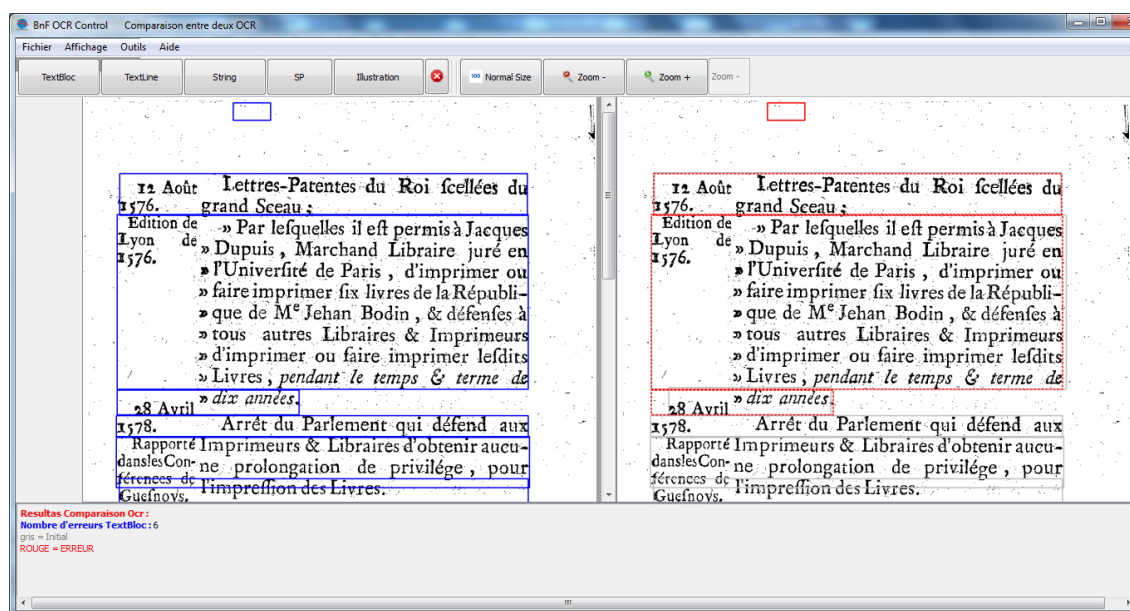


FIGURE 4.38 – Comparaison entre les TextBlocks de deux OCR (Etape 2)

→ Apports

Ce projet nous a été une occasion d'une part pour améliorer mes compétences en terme d'analyse, de conception, de développement et de gestion de projet multi-média et de travailler dans un cadre professionnel plus exigeant en terme de qualité de service et de temps de réalisation. Aussi ce projet m'a permis d'acquérir des connaissances techniques dans le domaine de traitement des documents numérique et de la reconnaissance optique de caractères.

Du point de vue gestion de projet, à travers ce projet, j'ai pu apprendre une nouvelle démarche de gestion de projet telle que la démarche des méthodes d'agile.

Au niveau conception j'ai amélioré durant le déroulement de la phase de conception mes compétences et mes connaissances en termes de conception orientée objet en utilisant la langage UML.

Finalement au niveau de développement , ce projet a été une bonne occasion pour moi d'enrichir mes connaissances en termes de programmation en utilisant la langage de programmation C++ et la bibliothèque d'interface graphique QT.

Conclusion générale

Tout au long de ce projet nous avons conçu et réalisé une application qui offre de nouveaux outils pour les contrôleurs de la BNF de bien vérifier les résultats de l'OCR. Cette application assure la correction de la segmentation et correction des strings, elle permet aussi de comparer deux OCR et recherche des mots.

D'une manière générale, notre application offre aux contrôleurs de la BNF des outils faciles et très ergonomiques qui lui permettent de faire une vérification et correction des résultats de l'OCR selon une nouvelle approche multimedia. Les contrôleurs de la BNF peuvent par exemple corriger l'objet String selon cote correction des mots et correction de la segmentation.

Par ailleurs, au terme de ce projet nous avons pu exploiter nos connaissances théorique et pratique en C++ et aussi enrichir nos connaissances sur les interfaces graphique QT et sur des nouvelles méthodes de travail comme les méthodes agiles.

L'intérêt principal que nous avons tiré de cette étude est que nous avons bien affronté la vie professionnelle de notre domaine. Nous avons évalué les différentes étapes de réalisation d'un projet ainsi que les techniques développés par les spécialistes du domaine pour assurer l'efficacité et la bonne réalisation des travaux en se limitant à des durées de temps exactes. Ainsi, nous avons pu voir la complexité de la mise en route d'un nouveau projet et sa rapide évolution qui nous a appris à nous mieux organiser afin d'être capable de finaliser notre travail.

Enfin, comme tout travail notre projet ne manque pas de perspectives. Pour cela nous avons prévu d'implémenter des extensions afin d'améliorer encore notre application. Ces extensions visent essentiellement l'intégration de nouvelles techniques basées sur le traitement d'images.

Bibliographie

- [1] Rémy Mullot, germes Lavoisier, 2006 France. *Les documents écrits de la numérisation à l'indexation par le contenu.*
- [2] Nicholas Journet and Rémy Mullot and Véronique Eglin and Jean-Yves Ramel, A. 2006. *Analyse d'images de documents anciens :catégorisation de contenus par approche texture* CIFED, Colloque International sur l'Ecrit et le Document Jouve - Paris.
- [3] Wahl F, Wong K. 1982. Computer graphics and image procesing *Block segmentation and text extraction in mixed text documents*, n °20p. 375-390.