

Rapport de Stage

Développement d'un démonstrateur de technologies
de reconnaissance de formulaires et de caractères
pour le projet Σ ureka! "RecoNomad".

Antoine MERCIER

Master 2^{ème} année

Mention Informatique, Mathématiques et leurs Applications

Parcours Professionnel Génie Informatique

Université de La Rochelle

Responsables scientifiques & pédagogiques :

Jean-Marc Ogier et Patrick Franco

Sous la direction de :

Jean-Marc Wallet

3 juillet 2008

Table des matières

Remerciements	3
Introduction	4
Glossaire	5
1 Contexte Professionnel	7
1.1 Σureka !	8
1.2 Oséo	8
1.3 ULR-Valor	8
1.4 L3i - Université de La Rochelle	9
1.5 DocLedge	9
2 Au cœur du projet RecoNomad	10
2.1 Contexte	10
2.2 Planification des tâches	11
3 Le Démonstrateur	12
3.1 Un réel besoin	12
3.2 Contexte et choix technologiques	12
3.3 Fonctionnalités développées	13
3.3.1 Affichage du signal de la tablette	14
3.3.2 Analyseur de syntaxe du fichier des résultats	15
3.3.3 Visualisation des temps de traitements	15
3.3.4 Visualisation de la segmentation	16
3.3.5 Visualisation et analyse des résultats de l'OCR	17
3.4 Analyse des résultats	20
3.4.1 Ambiguïté sur le choix du formulaire	20
3.4.2 La segmentation des informations	21
4 Les Signatures & Caractéristiques	22
4.1 Rappel sur la reconnaissance de caractères	22
4.1.1 Fonctionnement d'un OCR standard	22
4.1.2 Fonctionnement de notre OCR	23
4.2 Signatures	24
4.2.1 Contexte	24
4.2.2 Travail effectué	24

4.3	Les Caractéristiques	24
4.3.1	Contexte	24
4.3.2	Travail effectué	25
4.4	Analyse des résultats	25
5	Outils & contributions	26
5.1	Outils employés	26
5.2	Contributions	27
	Conclusion	28
	Bibliographie	30
	Annexes	31

Remerciements

Je souhaiterais remercier tout d'abord Jean-Marc Ogier de m'avoir accordé sa confiance et, plus globalement, toute l'équipe RecoNomad pour l'accueil et la confiance qu'ils m'ont accordée. Je remercie aussi beaucoup ULR-Valor pour m'avoir pris en stage et considéré comme l'un des leurs à part entière. Je remercie aussi les membres du L3i pour cette dernière raison.

Merci aux occupants de la salle 116 pour la bonne entente que nous avons eu durant ces quelques mois. Je pense particulièrement aux stagiaires grecs qui ont apporté un peu d'animation supplémentaire.

Merci aussi à l'équipe de la cellule @cTice et aux doctorants du L3i qui m'ont supporté lors des repas tout au long de ce stage.

Enfin, merci à Patrick Lacroix et Daniel Weets de l'entreprise DocLedge pour leur bonne humeur lors de leurs venues.

Introduction

Ce rapport a pour but de rendre compte de mon activité lors de mon stage de fin d'étude pour l'obtention de mon diplôme de Master. Malgré le fait que je me sois inscrit en parcours professionnel, j'ai effectué ce stage au sein d'une équipe de recherche du Laboratoire d'Informatique, Image et Interaction (L3i) à l'Université de La Rochelle. Ce choix fut délibéré et sera expliqué dans le premier chapitre de ce rapport.

Durant ce stage de 20 semaines, j'ai dû travailler seul, de concert avec l'équipe ou encore avec nos partenaires belges de l'entreprise DocLedge. J'ai eu des phases de développement pur, des phases de recherche et développement et aussi, de façon plus ponctuelle, des phases de recherche. Ce rapport sera donc découpé en différents chapitres dont le premier (p. 7) situera précisément l'environnement de travail dans lequel j'ai évolué. Les deuxième chapitre (p. 10) présentera l'équipe dans laquelle j'ai travaillé et la planification des tâches effectuées durant ce stage. Les autres chapitres (p. 12 et 22) concerneront chacune des parties ayant constitué mon sujet de stage et le dernier (p. 26) présentera les outils logiciels utilisés ainsi que les diverses contributions que j'ai pu apporter durant cette période.

L'essentiel de ce stage a eu pour but de concevoir et mettre en place une plate-forme de test des Dynamic Loading Libraries (DLLs) développées par l'équipe de recherche, lesquelles seront utilisées et commercialisées par l'entreprise DocLedge. Cette plate-forme sera aussi utilisée dans le futur par le L3i pour montrer son savoir, son savoir-faire et ses compétences lors de démonstrations.

Glossaire

API - Application Programming Interface

Interface de programmation d'applications, contenant un ensemble de fonctions courantes de bas niveau, bien documentées, permettant de programmer des applications de "Haut Niveau". On obtient ainsi des bibliothèques de routines.

DLL - Dynamic Loading Library

Il s'agit d'un type de bibliothèque de fonctions logicielles qui permet de n'être chargée en mémoire que si elle est nécessaire.

IHM - Interface Homme Machine

L'interface Homme-machine ou interaction humain-machine étudie la façon dont les humains interagissent avec les ordinateurs ou entre eux à l'aide d'ordinateurs, ainsi que la façon de concevoir des systèmes informatiques qui soient ergonomiques, c'est-à-dire efficaces, faciles à utiliser ou plus généralement adaptés à leur contexte d'utilisation.

JDK - Java Development Kit

Il s'agit de l'environnement dans lequel le code Java est compilé pour être transformé en "bytecode" afin que la JVM puisse l'interpréter.

JNI - Java Native Interface

Il s'agit d'un framework (soit un ensemble de bibliothèques, d'outils et de conventions facilitant le développement) qui permet à du code Java s'exécutant à l'intérieur de la JVM d'appeler et d'être appelé par des applications natives (c'est-à-dire des programmes spécifiques au matériel et au système d'exploitation de la plate-forme concernée), ou avec des bibliothèques logicielles basées sur d'autres langages (C, C++, assembleur...).

JVM - Java Virtual Machine

Ce programme est spécifique à chaque plate-forme ou couple (machine/système d'exploitation) et permet aux applications Java compilées en "bytecode" de produire les mêmes résultats quelle que soit la plate-forme, tant que celle-ci est pourvue de la machine virtuelle Java adéquate.

MFC - Microsoft Foundation Classes

Il s'agit d'une bibliothèque de classes en C++ encapsulant l'API Win32 (écrite en C) de Windows. Elle offre également un framework de développement de type Document/Vue inspirée du motif de conception Modèle Vue-Contrôleur (MVC).

MSDNAA - Microsoft Development Network Academic Alliance

Il s'agit du programme de licences Microsoft destiné aux établissements d'enseignement supérieur. Ce programme permet aux étudiants des établissements ayant souscrit à l'offre de pouvoir utiliser à titre gratuit différents logiciels de l'éditeur. L'ensemble des logiciels proposés est constitué notamment de systèmes d'exploitation, d'environnements de développement, de serveurs d'application et d'outils bureautiques.

OCR - Optical Character Recognition

Il s'agit d'un domaine encore ouvert de l'informatique consistant à mettre en oeuvre des méthodes de reconnaissance automatique de caractères numérisés. Cette numérisation est le plus souvent effectuée avec un scanner. Dans notre cas, le tracé des caractères est capté à l'aide du stylo et de la tablette électronique.

SGBD - Système de Gestion de Base de Données

Il s'agit d'un ensemble de programmes qui permet la gestion et l'accès à une base de données. Un SGDB héberge généralement plusieurs bases de données, qui sont destinées à des logiciels ou des thématiques différentes.

SVG - Scalable Vector Graphics

Il s'agit d'un format de données conçu pour décrire des ensembles de graphiques vectoriels et basé sur XML. Ce format est spécifié par le W3C. Il est né de la fusion des formats proposés par Microsoft (Vector Markup Language) et Adobe (Precision Graphics Markup Language).

W3C - World Wide Web Consortium

Il s'agit d'un organisme de normalisation fondé en octobre 1994 pour promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XHTML, XML, RDF, CSS, PNG, SVG et SOAP. Le W3C n'émet pas des normes au sens européen, mais des recommandations à valeur de standards industriels.

XML - eXtensible Markup Language

Il s'agit d'un langage informatique de balisage générique. Son objectif est de faciliter l'échange automatisé de contenus entre systèmes d'informations hétérogènes (d'où une interopérabilité).

Chapitre 1

Contexte Professionnel

L'une des particularités de ce stage est qu'il se situe dans un contexte collaboratif entre plusieurs entités. En effet, stagiaire d'ULR-Valor, j'ai travaillé de concert avec les ingénieurs de la société DocLedge et les chercheurs du L3i. Ces deux entités travaillent ensemble sur le projet RecoNomad labellisé par le programme européen Σ ureka! et financé en France par l'Oséo (cf. Fig. 1.1). Une présentation de chacun de ces organismes est donc nécessaire afin de comprendre le milieu dans lequel ce stage s'est déroulé.

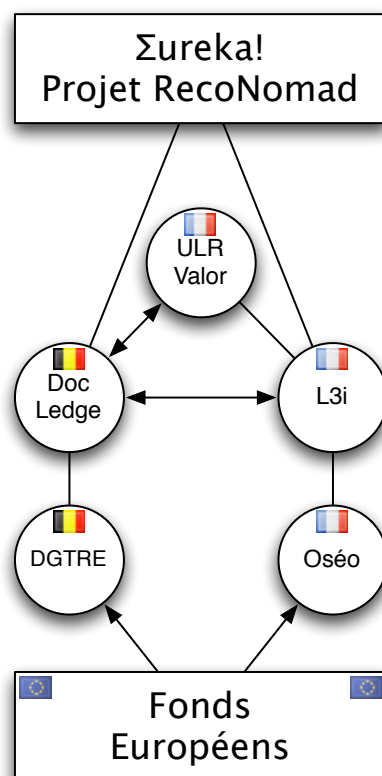


FIG. 1.1 – Relations et échanges au sein du programme Σ ureka!

1.1 Σureka !

Σureka! est une initiative intergouvernementale qui vise à renforcer la compétitivité de l'Europe par la promotion de la recherche et du développement transfrontaliers. Elle permet aux industriels et aux instituts de recherche de collaborer afin de développer et exploiter des technologies novatrices. En France, Σureka! est principalement financée par l'Oséo.

Dans le cas du projet RecoNomad, la partie recherche est effectuée par le L3i et la partie industrialisation est gérée par l'entreprise belge DocLedge. Du point de vue des ressources humaines, le L3i a recruté une jeune chercheuse et a délégué la partie développement à deux ingénieurs d'ULR-Valor. Du côté de DocLedge, il y a deux ingénieurs et deux commerciaux liés à ce projet.

1.2 Oséo

Oséo est un Établissement Public de l'État qui a pour mission de financer et accompagner les PME, en partenariat avec les banques et les organismes de capital-investissement, dans les phases les plus décisives du cycle de vie des entreprises (cf. Fig. 1.2).



FIG. 1.2 – Organisation juridique d'Oséo depuis 2007.

1.3 ULR-Valor

Cette entreprise est la filiale de l'Université de La Rochelle créée pour gérer efficacement les relations contractuelles entre les industriels et l'université. Ses missions sont d'être une interface entre l'université et le monde socio-économique, de soutenir les projets de valorisation de l'université, d'accompagner les porteurs de projet dans leurs démarches, de rédiger et négocier les contrats de recherche, de prestation et de transfert, de monter des dossiers de financements de projets, d'assurer le suivi de la protection des créations intellectuelles de l'université et d'accompagner les créateurs d'entreprises innovantes et les enseignants-chercheurs désirant collaborer avec des entreprises. Le L3i sous-traite la partie développement à ULR-Valor pour la réalisation du projet.

1.4 L3i - Université de La Rochelle

Le L3i est reconnu par le Ministère de la Recherche comme Équipe d'Accueil (EA2118) depuis 1996. Il regroupe désormais une quarantaine d'enseignant-chercheurs (dont 12 habilités à diriger des recherches), essentiellement issus de la communauté informatique (section 27 du CNU) et génie informatique (section 61) de l'Université de La Rochelle. Au premier janvier 2008, 28 doctorants y sont inscrits.

Ce laboratoire est subdivisé en trois axes de recherche : Sémantique et Intermédiation de Données (Sido), Images et Interactivité Numérique (ImagIN) et Image, Media numériques et Documents (I-MeDoc). Ce dernier est constitué de divers projets autour de la gestion de masses de documents.

L'équipe, qui fait partie de l'axe I-MeDoc, dans laquelle j'ai travaillé durant ces 20 semaines est dirigée par Jean-Marc Ogier, Professeur des Universités. Elle est liée par un partenariat avec l'entreprise belge DocLedge pour la réalisation du projet RecoNomad. Durant cette période, j'ai été installé dans les locaux du L3i et j'ai eu à ma disposition une station de travail performante (Intel Xeon double cœur 2GHz avec 2Go de mémoire vive).

1.5 DocLedge

L'entreprise DocLedge (anciennement e-Connector, cf. Annexe 1) située en Belgique commercialise des outils informatiques d'acquisition, de manipulation, de gestion et d'archivage de documents électroniques pour des entreprises souhaitant automatiser le fonctionnement de leur système d'information. Par exemple, les sociétés effectuant des enquêtes auprès d'un certain public ou encore des sociétés exploitant des données extraites de formulaires comme ceux d'un constat d'accident ou des caisses d'allocation familiale.

Chapitre 2

Au cœur du projet RecoNomad

2.1 Contexte

Certains des produits commercialisés par DocLedge sont des tablettes électroniques permettant l’acquisition d’écriture effectuée sur un bloc de papier standard. Celles-ci enregistrent donc le signal dit “en-ligne” correspondant au tracé fait sur la feuille située sur la tablette. De ce fait, l’utilisateur conserve son original sur papier et peut obtenir, en connectant la tablette sur son ordinateur, un fichier contenant les informations vectorielles du tracé effectué.

L’idée du projet RecoNomad est de développer un ensemble d’outils informatiques permettant de gérer le remplissage de formulaires papier sur ces tablettes. À terme, le processus souhaité est le suivant : l’utilisateur effectue une campagne de saisie de formulaire durant une journée de travail. À la fin de la journée, celui-ci connecte la tablette à son ordinateur personnel, récupère les documents générés par la tablette et “injecte” ces mêmes documents dans le système RecoNomad (cf. Fig. 2.1). À partir de ce moment là, le système RecoNomad va, de façon automatique, détecter quel type de formulaire a été rempli, associer les informations venant de la tablette à un formulaire modèle et enfin effectuer une reconnaissance des caractères. Le résultat de ce traitement pourra être indexé et archivé par un système auxiliaire pour que l’utilisateur puisse facilement retrouver et réutiliser ses documents par la suite.

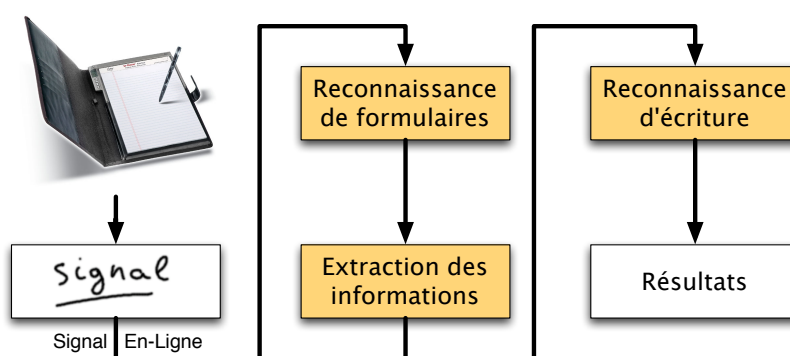


FIG. 2.1 – Synoptique représentant les différentes étapes de traitement d’un document.

Le partenaire DocLedge a établi comme contrainte que ce système soit développé sous la forme de briques logicielles (appelées “DLL” dans l’environnement Windows). En parallèle, il a été décidé qu’un logiciel à but scientifique et technique soit développé afin de pouvoir facilement observer les résultats de ces briques logicielles et d’en améliorer les performances. Ce logiciel de test, ne faisant qu’utiliser les DLLs, doit aussi servir dans le futur à montrer les compétences des membres du laboratoire L3i au sein de la communauté (aussi bien grand public que scientifique). Voilà pourquoi on l’appellera par la suite “démonstrateur”.

2.2 Planification des tâches

Le projet RecoNomad a débuté de façon concrète en septembre 2007 par l’arrivée de Laetitia Rousseau, alors jeune docteur. Son expérience dans le domaine des signaux “en-ligne” est un plus pour ce projet et c’est pourquoi son travail a été d’étudier l’ajout de caractéristiques “en-ligne” aux caractéristiques “hors-ligne” pour l’amélioration des résultats du système. Ensuite, Clément Rouzière et Jonathan Taquet l’ont rejointe afin de l’épauler dans le développement. Durant cette période, ils ont effectué les tâches d’analyse du problème et de rédaction des cahiers des charges fonctionnels et techniques. Par la suite, Clément Rouzière a étudié la problématique de la détection du formulaire utilisé et Jonathan Taquet a étudié la segmentation des champs, les formats de fichiers et la structure globale du système. Enfin, ensemble, ils ont étudié les différents systèmes de classification existants dans la littérature.

À mon arrivée dans l’équipe, une étude des divers documents rédigés a été nécessaire pour m’imprégner du travail déjà effectué. De cette étude sont ressortis deux éléments principaux sur lesquels je pouvais apporter ma contribution. Le premier travail concerne l’amélioration des signatures étudiées. La seconde tâche concerne le développement intégral du démonstrateur. Enfin, un complément du premier travail est nécessaire pour réorganiser l’utilisation des caractéristiques. Celle-ci doit avoir comme but de rendre cette partie du projet plus modulaire de façon à améliorer les temps de traitements de la chaîne lors de la reconnaissance des caractères. Ces différents travaux sont ponctués par les visites des partenaires belges, celle de Christian Clavaud de l’Oséo ainsi que de présentations devant certains membres du laboratoire.

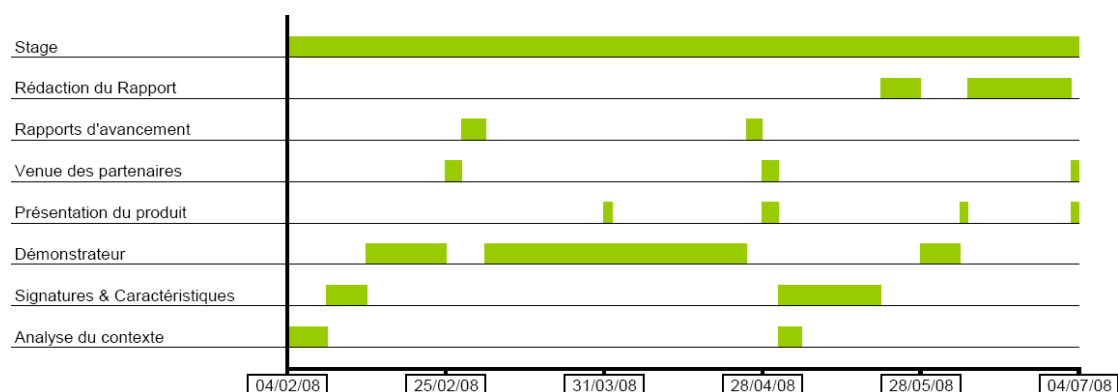


FIG. 2.2 – Diagramme de Gantt représentant l’aménagement du temps durant le stage.

Chapitre 3

Le Démonstrateur

3.1 Un réel besoin

Lorsque le projet a démarré, très rapidement l'idée d'un démonstrateur est apparue. En effet, tout au long de ce projet prévu sur deux années, diverses présentations sont et seront effectuées. Naturellement, la plupart d'entre elles sont effectuées auprès des partenaires belges mais aussi, de façon plus ponctuelle, auprès des représentants d'Oséo. Bien que l'utilisation première de ce démonstrateur soit celle de la correction de bugs présents dans les DLLs qui constituent le coeur même du système RecoNomad, il servira aussi à des présentations effectuées au sein même du laboratoire ou encore lors de séminaires effectués dans d'autres universités ou auprès de futurs étudiants intéressés par les domaines de recherche du laboratoire.

3.2 Contexte et choix technologiques

Lors de mon arrivée dans l'équipe, j'ai dû me contraindre aux exigences de développement établies par nos partenaires belges. Les DLLs devant être écrites en C++ sous environnement Windows, il paraissait logique de créer l'interface graphique les accompagnant en C++ en utilisant les Application Programming Interfaces (API) Win32 ou Microsoft Foundation Classes (MFC).

Cependant, divers problèmes se sont posés. Développer en MFC (ou Win32) requerrait que j'utilise l'outil de développement Microsoft Visual Studio Professionnel. Bien qu'installé sur nos ordinateurs de travail, nous ne pouvions l'utiliser pour produire quoi que ce soit à but commercial (DLLs incluses) en raison de la licence Microsoft Development Network Academic Alliance (MSDNAA) restrictive à ce sujet. Nous avons donc dû utiliser Microsoft Visual Studio Express qui est une version fournie gratuitement par Microsoft mais qui n'inclut pas les outils de développement d'interface graphique. Par ailleurs, mon expérience avec les MFC étant faible, je ne pouvais pas décemment re-découvrir ces APIs dans le temps qui m'était imparti (la première version exploitable devait être prête pour la fin du mois de février, c'est à dire en moins d'un mois).

Tout ceci m'a conduit à décider de développer le démonstrateur en Java. En effet, tous les outils de développement sont gratuits et ne sont pas soumis aux mêmes restrictions que les outils de Microsoft. Par ailleurs, pour effectuer les échanges d'informations entre les

DLLs et l'interface homme machine (IHM), j'ai étudié le fonctionnement des Java Native Interfaces (JNI) et j'ai jugé que la quantité de travail qu'implique leur utilisation n'était pas justifiée. De plus, l'équipe avait déjà mis au point un petit programme écrit en C++ qui exploitait les DLLs. Celui-ci s'exécutait en ligne de commande et générait un fichier eXtensible Markup Language (XML) contenant les résultats des différents traitements. À partir de là, mon travail se résumait alors à faire une interface graphique permettant d'exécuter le dit programme pour fournir une représentation graphique du contenu du fichier généré. Du fait de la nature multi-plateforme du Java, celle-ci est utilisable sur tout système ayant une Java Virtual Machine (JVM) d'installée mais l'utilisation des DLLs, elle, est réservée aux plates-formes Windows.

Il était convenu que tous les deux mois une rencontre aurait lieu entre les deux équipes (L3i et DocLedge). La première venue des partenaires après le début de mon stage étant fin février, il imposait que la première ébauche du démonstrateur devait être disponible à ce moment là et qu'une version quasi-définitive soit prête pour la fin du mois d'avril. De plus, il m'a été demandé de préparer une démonstration interne au laboratoire vers la fin du mois de mars. Celle-ci avait pour but de montrer mon avancement et les fonctionnalités mises en place, ainsi que d'obtenir éventuellement des critiques constructives venant de personnes ne suivant pas mes travaux au jour le jour. C'est pourquoi, j'ai décidé que mon planning suivrait cette cadence et que je devais donc avoir une version exploitable à la fin de chaque mois.

3.3 Fonctionnalités développées

Ce démonstrateur avait ainsi deux buts d'égale importance : la démonstration d'un savoir-faire technologique et l'aide au développement de ces technologies. Pour cela, il devait permettre de quantifier aussi bien visuellement que numériquement les résultats des DLLs et ce, de la façon la plus exhaustive et rapide possible. La difficulté de ce travail venait de la présence de nombreux facteurs à contrôler. En effet, le processus de reconnaissance d'un formulaire contient plusieurs étapes qui se subdivisent en sous-étapes et pour chacune, des paramètres sont pris en considération pour optimiser le fonctionnement du système (cf. Fig. 3.1).

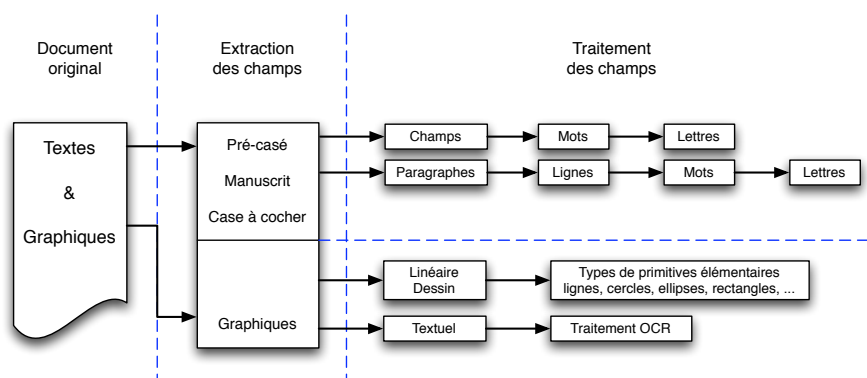
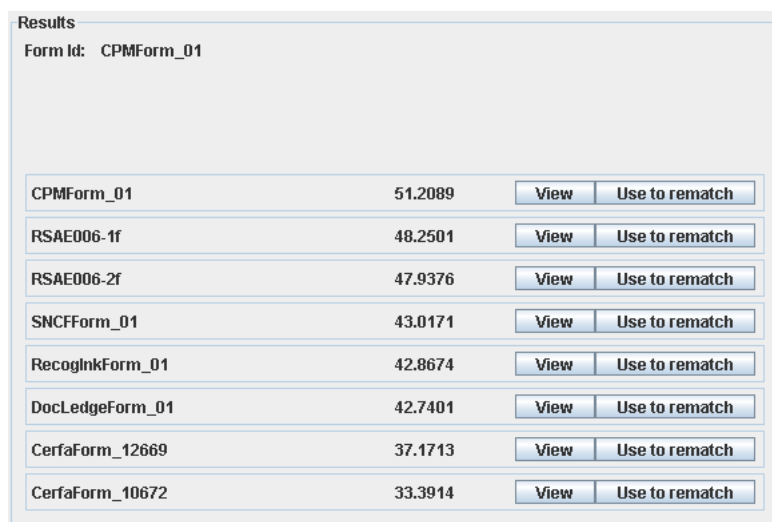


FIG. 3.1 – Synoptique représentant les tâches effectuées par les outils développés.

3.3.1 Affichage du signal de la tablette

La détection du type de formulaire est basée sur des méthodes de mesure de densité de remplissage du document. C'est pourquoi lors de cette détection, nous avons comme réponse de la DLL un taux de confiance pour chaque formulaire présent dans la base de données. Ce taux, exprimé en pourcentage, indique donc le degré de certitude du système quant à la correspondance entre le document à traiter et chacun des formulaires vierges présents dans notre base de données. D'après les tests que nous avons effectués, lorsque le système se trompe de formulaire, c'est en fait qu'il y a deux formulaires ayant des taux similaires et que le bon a un taux de confiance légèrement inférieur à l'autre. Le processus de décision est externe à cette DLL et pour nos tests, nous avons choisi de prendre le premier de la base ayant le taux de confiance maximum. Cette prise de décision pourrait être pondérée, par des informations statistiques de fréquence d'utilisation des formulaires par exemple, ce qui permettrait d'éviter les rejets ou les ambiguïtés. Ceci n'étant pas de notre ressort, le démonstrateur permet d'obtenir ces taux de confiance et de relancer le processus de reconnaissance en ciblant le formulaire utilisé dans le cas d'une ambiguïté (cf. Fig. 3.2).



Results		
Form Id: CPMForm_01		
CPMForm_01	51.2089	View Use to rematch
RSAE006-1f	48.2501	View Use to rematch
RSAE006-2f	47.9376	View Use to rematch
SNCFForm_01	43.0171	View Use to rematch
RecoglnkForm_01	42.8674	View Use to rematch
DocLedgeForm_01	42.7401	View Use to rematch
CerfaForm_12669	37.1713	View Use to rematch
CerfaForm_10672	33.3914	View Use to rematch

FIG. 3.2 – Zone où les taux de confiances sont affichés.

Afin de contrôler le résultat de la détection du type de formulaire, il fallait pouvoir voir et comparer le signal d'origine venant de la tablette, le signal recalé, le masque ayant servi au recalage et plus accessoirement le formulaire original vierge. En effet, comme le signal venant de la tablette n'est pas tout à fait au format A4 standard de 210×297 mm, il a fallu mettre au point une méthode de recalage des données sources. Par ailleurs, comme nous ne contrôlons pas le format du document rempli, un effet d'échelle peut être présent. C'est pourquoi, dans la méthode de recalage, a aussi été intégrée cette notion d'homothétie. Pour aider au développement, à l'amélioration de la méthode, mais aussi à la compréhension des erreurs commises par le système, j'ai dû intégrer cette visualisation au démonstrateur. C'est ainsi que dès les premières versions du démonstrateur, celui-ci a eu comme fonctionnalité de pouvoir afficher le signal original, le formulaire original et le masque employé (cf. Fig. 3.3).

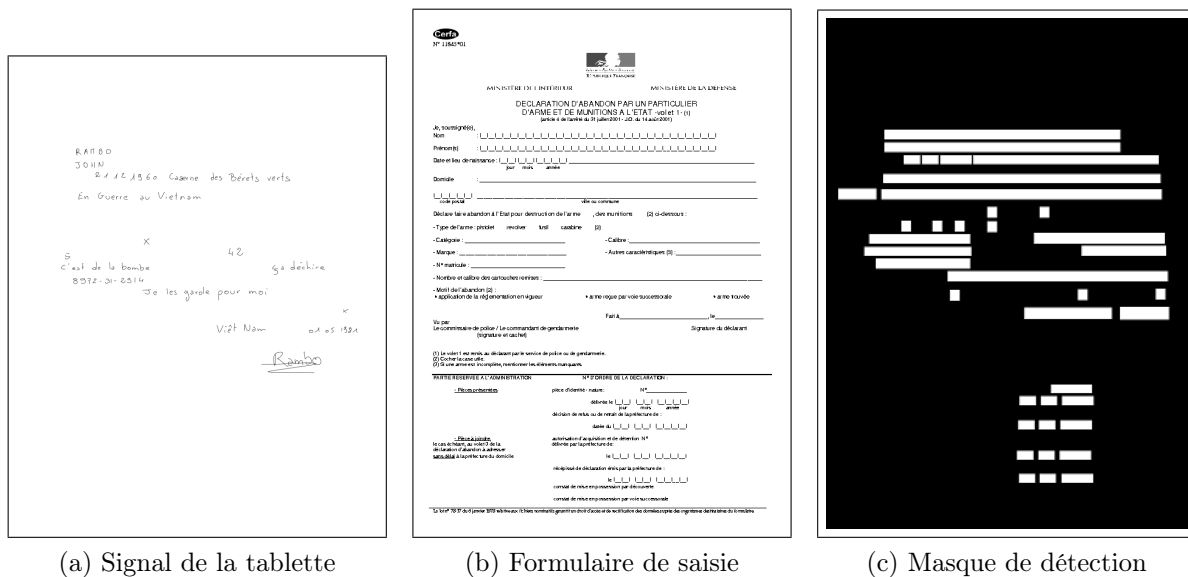


FIG. 3.3 – Informations visuelles affichées.

En un premier temps, cette visualisation s’effectuait sous la forme de trois onglets différents (cf. Annexe 5.2). Actuellement, celle-ci s’effectue sous la forme de calques superposables en cliquant sur les boutons correspondant (cf. Annexe 5.3).

3.3.2 Analyseur de syntaxe du fichier des résultats

À l’utilisation des DLLs, un fichier de résultat est généré. Celui-ci est dans un format que nous avons nous-mêmes défini à l’aide du langage XML et contient presque toutes les informations résultantes des différents traitements effectués sur le document venant de la tablette. Les quelques informations qui ne sont pas dans ce fichier ont été volontairement mises de côté car nous ne voulions pas surcharger le format que nous avons défini. Ces informations sont principalement les taux de reconnaissance du formulaire et les temps de calculs. Ces informations sont toutefois disponibles en interfaçant les sorties standard et erreur des DLLs. Comme le fichier généré est dans un format défini en XML, j’ai dû comprendre le fonctionnement de l’analyseur XML intégré au langage Java afin de pouvoir en extraire les informations à afficher. Dans le but de simplifier certains traitements, j’ai recréé en mémoire l’arbre correspondant au contenu de ce document. Cet arbre, basé sur des classes héritées de `DefaultMutableTreeNode`, permet d’afficher le document sous la forme d’une arborescence ressemblant à un explorateur de fichier. Ceci permet d’affecter un comportement différent sur chaque type de noeud. Par ailleurs, cette hiérarchie de classe est doublée d’une autre hiérarchie identique correspondant aux “Panels” permettant de visualiser les informations rattachées au noeud cliqué.

3.3.3 Visualisation des temps de traitements

Comme dit précédemment, pour obtenir les temps de traitement, il suffit de s’interfacer avec la sortie erreur. En effet, il nous a paru plus simple de faire ressortir ces informations

par ce biais-là car cela permet de les obtenir très simplement si on exécute le programme au travers de la ligne de commande. Il a été choisi d'afficher ces temps de traitement dans une fenêtre séparée (cf. Fig. 3.4) afin de ne pas alourdir la fenêtre principale. Dans cette petite fenêtre, nous avons donc une partie textuelle rafraîchie au fur et à mesure de l'avancement des tâches, une barre de progression ainsi qu'un bouton pour annuler le traitement qui se transforme en bouton de fermeture de la fenêtre à la fin du processus.

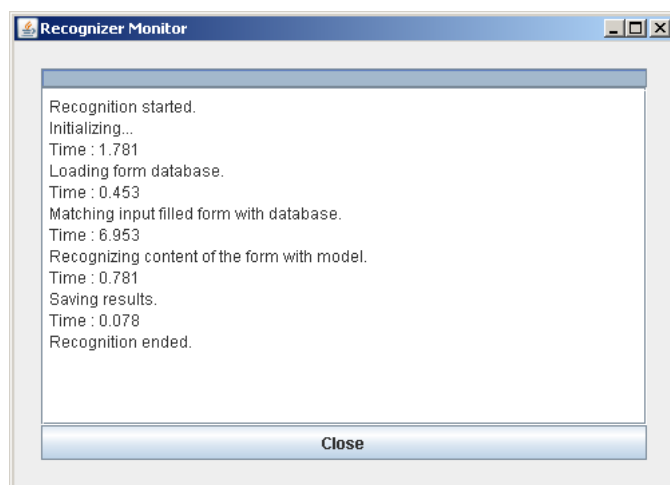


FIG. 3.4 – Fenêtre affichant les temps de traitement.

3.3.4 Visualisation de la segmentation

Dans le processus global de reconnaissance d'un formulaire, après la détection du type de formulaire rempli, il est nécessaire de segmenter le document recalé de façon à affecter les informations aux champs dans lesquels elles ont été supposées être écrites. Cela signifie que d'un document qui contient un ensemble non structuré d'informations, nous allons obtenir un document ayant une première structuration sous la forme de champs remplis. Une fois cette première structuration faite, une seconde est effectuée dans le cas de champs précasés. Ceci a pour effet, non seulement d'affecter une information à un champ mais en plus, d'affecter une lettre à une case. Dans le même temps, un recalage local est effectué afin de s'assurer que la lettre est bien dans la case après le premier recalage et l'éventuel redimensionnement. Ici aussi, le démonstrateur a eu un impact sur la rapidité de l'équipe à identifier les problèmes puisqu'avec celui-ci, nous pouvons afficher sous forme de calques le modèle de référence et le résultat du recalage (global et local) et de la segmentation. Les exemples suivants montrent l'information telle qu'elle est captée par la tablette avec en superposition le formulaire, ce qui permet de constater la nécessité de recalibrer les données (cf. Fig. 3.5a) et des cas typiques de mauvais recalage (cf. Fig. 3.5b) parce que le formulaire a été mal détecté et un bon recalage (cf. Fig. 3.5c) dans le cas où la détection du type de formulaire s'est correctement déroulée.

Demande de prime de déménagement

Merci de compléter soigneusement
un formulaire de déclaration de situation.

Sur 1.142.015 de validité de la Sécurité sociale
Après 1.142.015 de validité de la Sécurité sociale

► **Merci de rappeler votre identité**

Votre nom : **FOURC** Votre prénom : **PICKEC**

Votre date de naissance : 03/05/1974 (jj/mm/aa)

Votre adresse : 11061006

Code de département : 11061

Avancez-vous déjà d'autres aides (aides) pour payer votre déménagement ?

Si oui, indiquez le montant global des aides reçues :

Indiquez le nom et l'adresse des établissements ou organismes que vous avez accueilli ces aides :

► **Déclaration sur l'Éligibilité**

Je soussigné (Je soussignée) déclare sous serment que les données de prime au logement et de location sont exactes.

Je déclare avoir signé ce document en présence d'un officier public ou d'un officier de l'administration.

Le 11/06/2007, j'ai signé ce document en présence d'un officier public ou d'un officier de l'administration.

► **Pièces à joindre à votre demande**

► voir la liste des pièces à joindre de l'annexe de la prime au logement et de location ;

► voir les modalités des dépenses engagées pour votre déménagement (forme de location de résidence, forme d'acquisition, ...)

S 7102b

Emploie-moi réservé à la Cef
W 14000105 - Demande du 10/11/2007
DFO

(a) Avant recalage

Demande de prime de déménagement

Merci de compléter soigneusement
un formulaire de déclaration de situation.

Sur 1.142.015 de validité de la Sécurité sociale
Après 1.142.015 de validité de la Sécurité sociale

► **Merci de rappeler votre identité**

Votre nom : **FOURC** Votre prénom : **PICKEC**

Votre date de naissance : 03/05/1974 (jj/mm/aa)

Votre adresse : 11061006

Code de département : 11061

Avancez-vous déjà d'autres aides (aides) pour payer votre déménagement ?

Si oui, indiquez le montant global des aides reçues :

Indiquez le nom et l'adresse des établissements ou organismes que vous avez accueilli ces aides :

► **Déclaration sur l'Éligibilité**

Je soussigné (Je soussignée) déclare sous serment que les données de prime au logement et de location sont exactes.

Je déclare avoir signé ce document en présence d'un officier public ou d'un officier de l'administration.

Le 11/06/2007, j'ai signé ce document en présence d'un officier public ou d'un officier de l'administration.

► **Pièces à joindre à votre demande**

► voir la liste des pièces à joindre de l'annexe de la prime au logement et de location ;

► voir les modalités des dépenses engagées pour votre déménagement (forme de location de résidence, forme d'acquisition, ...)

S 7102b

Emploie-moi réservé à la Cef
W 14000105 - Demande du 10/11/2007
DFO

(b) Mauvais recalage

Demande de prime de déménagement

Merci de compléter soigneusement
un formulaire de déclaration de situation.

Sur 1.142.015 de validité de la Sécurité sociale
Après 1.142.015 de validité de la Sécurité sociale

► **Merci de rappeler votre identité**

Votre nom : **FOURC** Votre prénom : **PICKEC**

Votre date de naissance : 03/05/1974 (jj/mm/aa)

Votre adresse : 11061006

Code de département : 11061

Avancez-vous déjà d'autres aides (aides) pour payer votre déménagement ?

Si oui, indiquez le montant global des aides reçues :

Indiquez le nom et l'adresse des établissements ou organismes que vous avez accueilli ces aides :

► **Déclaration sur l'Éligibilité**

Je soussigné (Je soussignée) déclare sous serment que les données de prime au logement et de location sont exactes.

Je déclare avoir signé ce document en présence d'un officier public ou d'un officier de l'administration.

Le 11/06/2007, j'ai signé ce document en présence d'un officier public ou d'un officier de l'administration.

► **Pièces à joindre à votre demande**

► voir la liste des pièces à joindre de l'annexe de la prime au logement et de location ;

► voir les modalités des dépenses engagées pour votre déménagement (forme de location de résidence, forme d'acquisition, ...)

S 7102b

Emploie-moi réservé à la Cef
W 14000105 - Demande du 10/11/2007
DFO

(c) Bon recalage

FIG. 3.5 – Exemples d’affichage de recalage.

3.3.5 Visualisation et analyse des résultats de l’OCR

La dernière étape du processus de reconnaissance consiste à analyser et reconnaître les caractères qui ont été écrits dans les cases par l'utilisateur. Dans le démonstrateur, différentes vues sont disponibles pour apprécier ce résultat.

Noeud “Suggest” & encart “Results” (cf. Fig. 3.6)

Le premier système mis en place consiste en l’affichage du caractère correspondant à l’élément “Suggest” sélectionné dans l’encart “Results” de la fenêtre principale. Cet affichage contient aussi l’indice de confiance dans le caractère associé.

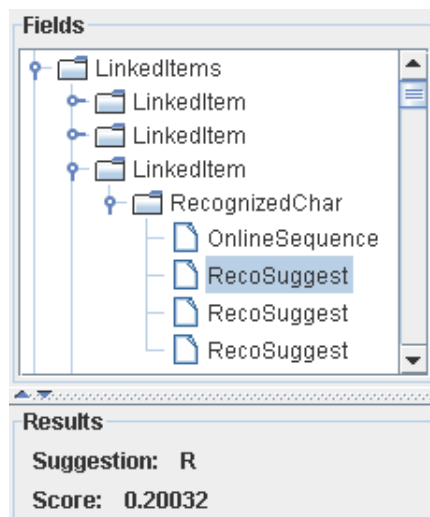


FIG. 3.6 – Zone affichant une suggestion de l’OCR et une partie de l’arborescence associée.

Le calque “OCR Results” (cf. Fig. 3.7)

Le deuxième système mis en place est une vue au travers du système de calque déjà présent. En effet, lors de l’appui sur le bouton “OCR Results”, un nouveau calque apparaît affichant le texte reconnu. Dans le cas où l’Optical Character Recognition (OCR) a retourné plus d’un résultat, celui ayant le meilleur score est choisi et en cas d’égalité, le premier alphabétiquement parlant est choisi. Cette méthode est très visuelle étant donné que le texte reconnu apparaît exactement là où il a été saisi par l’utilisateur. En effet, comme chaque caractère est traité indépendamment des autres, il est affiché dans la case correspondante. Pour cela, il a fallu régler un problème qui s’est posé par rapport à la gestion du texte en Scalable Vector Graphics (SVG). Dans ce langage, un rectangle est défini par son angle haut-gauche tandis que le texte est défini selon le point de départ de celui-ci sur l’axe de la “baseline”. Cette “baseline” est la ligne sur laquelle viennent se “poser” les caractères, ce qui donne le point de départ d’un caractère en bas à gauche de celui-ci. Pour régler ce problème, j’ai dû me résoudre à parcourir en parallèle l’arborescence du fichier de résultat et celle du fichier du modèle de formulaire et effectuer des jointures en me basant sur les identifiants des champs pour obtenir les coordonnées adéquates.

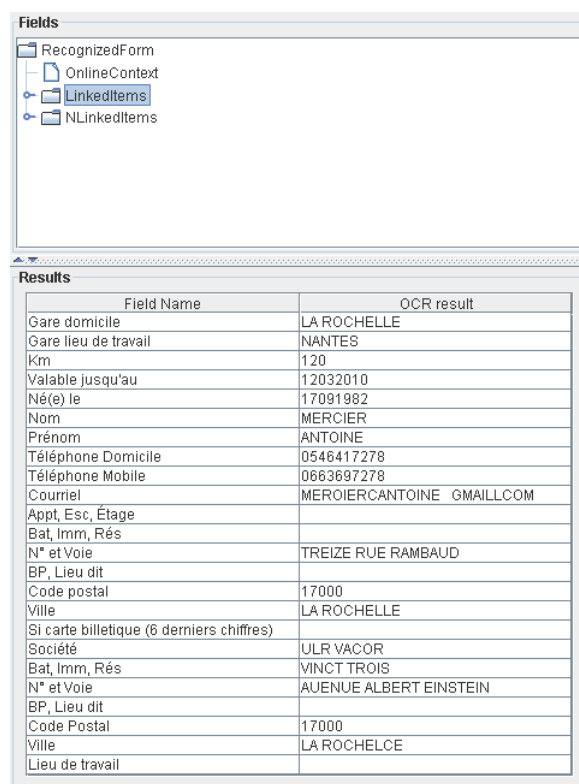
The image shows a form with various input fields. The OCR results are overlaid on the form, showing the recognized text in green. The form fields are as follows:

- Top right: [Empty box]
- First line: L A R O C H E L L E [Empty box]
- Second line: N A N T E S [Empty box]
- Third line: 1 2 0 1 2 0 3 2 0 1 0
- Fourth line: [Empty box]
- Fifth line: [Empty box] [Empty box] [Empty box] 1 7 0 9 1 9 8 2
- Sixth line: M E R C I E R [Empty box]
- Seventh line: A N T O I N E [Empty box]
- Eighth line: 0 5 4 6 4 1 7 2 7 8 0 6 6 3 6 9 7 2 7 8
- Ninth line: M E R C I E R C A N T O I N E [Empty box] G M A I L L E C O M [Empty box]
- Tenth line: [Empty box]
- Eleventh line: T R E I Z E R U E R A M B A U D [Empty box]
- Twelfth line: [Empty box]
- Thirteenth line: 1 7 0 0 0 L A R O C H E L L E [Empty box]
- Fourteenth line: [Empty box]
- Fifteenth line: [Empty box]
- Sixteenth line: [Empty box]
- Seventeenth line: [Empty box]
- Eighteenth line: [Empty box]
- Nineteenth line: [Empty box]
- Twentieth line: [Empty box]
- Twenty-first line: [Empty box]
- Twenty-second line: [Empty box]
- Twenty-third line: [Empty box]
- Twenty-fourth line: [Empty box]
- Twenty-fifth line: [Empty box]
- Twenty-sixth line: [Empty box]
- Twenty-seventh line: [Empty box]
- Twenty-eighth line: [Empty box]
- Twenty-ninth line: [Empty box]
- Thirtieth line: [Empty box]
- Thirty-first line: [Empty box]
- Thirty-second line: [Empty box]
- Thirty-third line: [Empty box]
- Thirty-fourth line: [Empty box]
- Thirty-fifth line: [Empty box]
- Thirty-sixth line: [Empty box]
- Thirty-seventh line: [Empty box]
- Thirty-eighth line: [Empty box]
- Thirty-ninth line: [Empty box]
- Fortieth line: [Empty box]
- Forty-first line: [Empty box]
- Forty-second line: [Empty box]
- Forty-third line: [Empty box]
- Forty-fourth line: [Empty box]
- Forty-fifth line: [Empty box]
- Forty-sixth line: [Empty box]
- Forty-seventh line: [Empty box]
- Forty-eighth line: [Empty box]
- Forty-ninth line: [Empty box]
- Fiftieth line: [Empty box]
- Fifty-first line: [Empty box]
- Fifty-second line: [Empty box]
- Fifty-third line: [Empty box]
- Fifty-fourth line: [Empty box]
- Fifty-fifth line: [Empty box]
- Fifty-sixth line: [Empty box]
- Fifty-seventh line: [Empty box]
- Fifty-eighth line: [Empty box]
- Fifty-ninth line: [Empty box]
- Sixtieth line: [Empty box]
- Sixty-first line: [Empty box]
- Sixty-second line: [Empty box]
- Sixty-third line: [Empty box]
- Sixty-fourth line: [Empty box]
- Sixty-fifth line: [Empty box]
- Sixty-sixth line: [Empty box]
- Sixty-seventh line: [Empty box]
- Sixty-eighth line: [Empty box]
- Sixty-ninth line: [Empty box]
- Seventieth line: [Empty box]
- Seventy-first line: [Empty box]
- Seventy-second line: [Empty box]
- Seventy-third line: [Empty box]
- Seventy-fourth line: [Empty box]
- Seventy-fifth line: [Empty box]
- Seventy-sixth line: [Empty box]
- Seventy-seventh line: [Empty box]
- Seventy-eighth line: [Empty box]
- Seventy-ninth line: [Empty box]
- Eightieth line: [Empty box]
- Eighty-first line: [Empty box]
- Eighty-second line: [Empty box]
- Eighty-third line: [Empty box]
- Eighty-fourth line: [Empty box]
- Eighty-fifth line: [Empty box]
- Eighty-sixth line: [Empty box]
- Eighty-seventh line: [Empty box]
- Eighty-eighth line: [Empty box]
- Eighty-ninth line: [Empty box]
- Ninetieth line: [Empty box]
- One hundred line: [Empty box]

FIG. 3.7 – Calque affichant le résultat de l’OCR superposé au masque du formulaire.

Noeud “LinkedItems” & tableau (cf. Fig. 3.8)

Le troisième système mis en place consiste à afficher un tableau contenant les intitulés des champs dans une colonne et le résultat de l’OCR dans une autre. Ce tableau apparaît lors de la sélection du noeud “LinkedItems” dans l’arbre du fichier de résultats.



The screenshot shows a software interface with two main sections. The top section, titled 'Fields', contains a tree view with the following items: 'RecognizedForm', 'OnlineContext', 'LinkedItems' (which is selected and highlighted in blue), and 'NLinkedItems'. The bottom section, titled 'Results', contains a table with two columns: 'Field Name' and 'OCR result'.

Field Name	OCR result
Gare domicile	LA ROCHELLE
Gare lieu de travail	NANTES
Km	120
Valable jusqu'au	12032010
Né(e) le	17091982
Nom	MERCIER
Prénom	ANTOINE
Téléphone Domicile	0546417278
Téléphone Mobile	0663697278
Courriel	MEROIERCANTOINE_GMAIL.COM
Appt, Esc, Étage	
Bat, Imm, Rés	
N° et Voie	TREIZE RUE RAMBAUD
BP, Lieu dit	
Code postal	17000
Ville	LA ROCHELLE
Si carte billettique (6 derniers chiffres)	
Société	ULR VACOR
Bat, Imm, Rés	VINCT TROIS
N° et Voie	AVENUE ALBERT EINSTEIN
BP, Lieu dit	
Code Postal	17000
Ville	LA ROCHELCE
Lieu de travail	

FIG. 3.8 – Zone affichant le tableau correspondant aux “LinkedItems”.

Menu contextuel (cf. Fig. 3.9)

Enfin, le dernier système mis en place consiste en l’affichage d’un menu contextuel lors du clic sur un caractère sur le calque affichant le tracé recalé. Ce menu affiche toutes les propositions avec leurs taux de confiance associés. Si le système n’a pas été en mesure de prendre une décision, le menu affiche alors “No suggestion”.

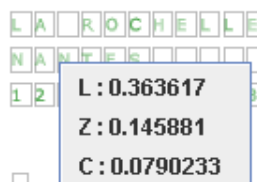


FIG. 3.9 – Menu contextuel affichant les suggestions de l’OCR sur le caractère “L”.

Le double clic

Afin de simplifier la phase d'analyse des résultats, un système de double clic sur le calque est disponible. Celui-ci fait refermer tous les noeuds de l'arbre du fichier des résultats et déployer celui correspondant au caractère sur lequel le double clic a été effectué. Cette fonction a été rajoutée pour simplifier la recherche d'un caractère précis lors de l'analyse des résultats sur un formulaire contenant un grand nombre de cases. En effet, dans les formulaires les plus précasés que nous avons essayés, il était facile de dépasser les 400 cases.

3.4 Analyse des résultats

3.4.1 Ambiguïté sur le choix du formulaire

Avec le démonstrateur, nous avons pu isoler les formulaires résistant à la reconnaissance. Il s'agit de ceux qui ont des structures similaires ou qui ont une structure particulière (densité de zones de saisie très faible ou très élevée, par exemple). En effet, avec la méthode mise en place, nous calculons un taux de confiance basé lui-même sur le calcul de critères. Le premier est une mesure de similarité entre deux densités, celle du modèle de formulaire avec pondération de la densité par la taille des champs et celle du formulaire rempli. En l'occurrence, il s'agit de la distance de Bhattacharyya qui est utilisé pour ce critère. Le second critère est un taux de remplissage qui correspond au pourcentage de signal en ligne se trouvant dans une zone de saisie du modèle. La pondération effectuée sur le premier critère permet d'éviter que les grandes zones de saisie masquent la réponse du calcul effectué sur les plus petites. Par contre, lorsque deux formulaires ont des structures assez similaires, leurs taux de confiance se retrouvent très proches (moins de 5%), et cela risque d'induire le système en erreur. Sans le démonstrateur, toutes ces observations auraient été difficiles à effectuer. L'utilisation du démonstrateur nous a tout de même fait constater que cette méthode est sensible dès lors que plusieurs formulaires ont sensiblement la même structure (cf. 3.10). Par ailleurs, il nous a permis de gagner du temps puisque nous pouvons voir dans celui-ci les taux de confiance du processus de détection du formulaire.

Cette méthode a comme inconvénient d'être indécise dans le cas de formulaires peu remplis. En effet, dans un tel cas, le peu d'information présente va induire un taux de confiance élevé sur les quelques cases contenant du texte mais du fait de la faible présence du signal, beaucoup de formulaires pourront avoir un taux de confiance similaire. En effet, si entre deux formulaires des champs se trouvent aux mêmes endroits et que seuls ces endroits sont remplis par le signal en ligne ou que le signal en ligne se trouve dans des zones vides sur ces deux formulaires, alors le taux de remplissage sera le même. Ceci fait que même si deux formulaires n'ont pas de structure globalement identique, ils peuvent tout de même être localement ambiguë. Ce cas de figure nous a été facile à détecter avec le démonstrateur puisque nous pouvions voir le signal et le formulaire détecté et comparer par rapport à la feuille effectivement remplie. Comme le démonstrateur permet de relancer le processus en spécifiant le bon formulaire, nous avons pu continuer nos autres tests sans avoir à modifier notre base de formulaire. Là encore, le gain de temps apporté par l'utilisation du démonstrateur a été important.

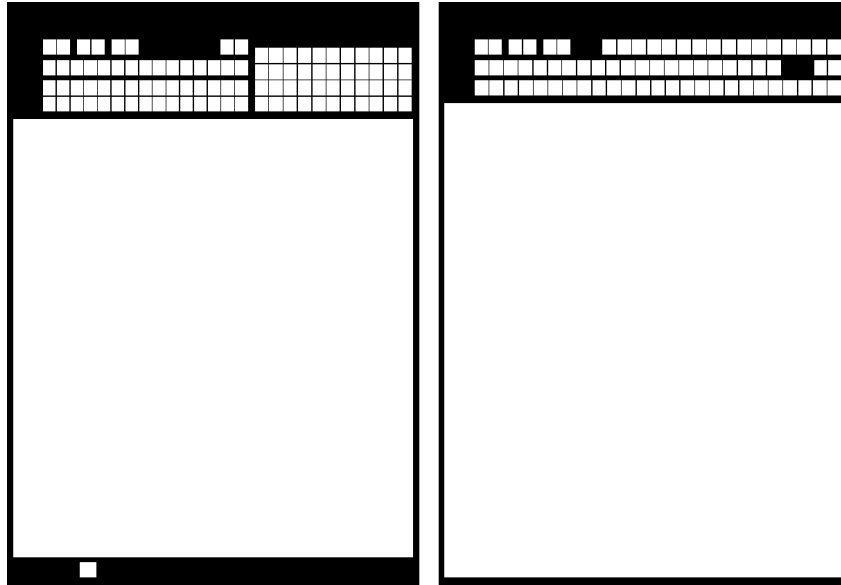
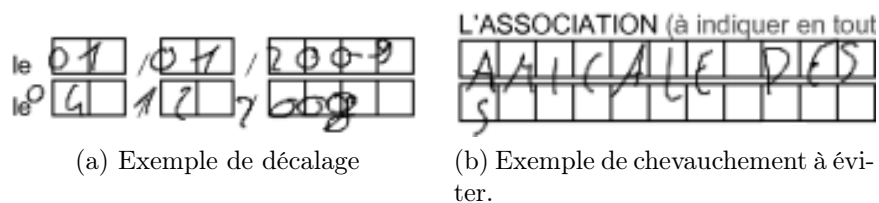


FIG. 3.10 – Exemples de formulaires structurés de façon similaire.

3.4.2 La segmentation des informations

Une fois que le formulaire a bien été détecté, une phase de recalage est nécessaire afin de pouvoir segmenter correctement les données saisies. En effet, comme nous avons pu le voir auparavant, cette phase de segmentation extrait les données et les lie aux zones de saisies définies dans le modèle de formulaire présent dans la base. Afin de lier les données correctement, il faut s'assurer qu'elles soient superposables aux cases correspondantes lors de la saisie (cf. Fig. 3.11a). De plus, ce traitement s'assure que les données ne chevauchent pas plusieurs zones de saisie (cf. Fig. 3.11b), auquel cas, la segmentation serait mal effectuée. Pour résoudre cette problématique, un premier recalage global est effectué de l'ensemble des données. Un second est effectué localement de façon à ce que le signal soit affecté à la bonne zone de saisie. De plus, celui-ci permet de pallier aux légers décalages qu'aurait pu subir la feuille durant la saisie. Dans le cas d'une zone pré-casé, les données sont systématiquement recalées vers la gauche de façon à ce que la première lettre du signal soit affectée à la première case de la zone de saisie. Le démonstrateur a servi ici pour visualiser le résultat de tous ces traitements et s'assurer du bon fonctionnement du système de segmentation des données.



(a) Exemple de décalage

(b) Exemple de chevauchement à éviter.

FIG. 3.11 – Exemples de problèmes induisant une mauvaise segmentation.

Chapitre 4

Les Signatures & Caractéristiques

4.1 Rappel sur la reconnaissance de caractères

4.1.1 Fonctionnement d'un OCR standard

La problématique de la reconnaissance de caractères fait partie des problématiques “ouvertes” (non résolues). En effet, il n'existe pas de solution exacte pour pouvoir reconnaître de l'écriture manuscrite de façon automatique. Jusqu'à maintenant, les méthodes les plus utilisées sont celles s'appuyant sur un système d'apprentissage de caractères provenant d'un maximum de scripteurs afin d'avoir une connaissance la plus large possible pour obtenir le taux de reconnaissance le plus élevé possible. Cette procédure d'apprentissage consiste à effectuer toute une série de traitements sur les caractères à apprendre. Une fois ces traitements effectués, nous obtenons une base nous permettant de reconnaître des caractères en appliquant à ces derniers les mêmes traitements qu'à ceux appris. Ces traitements ont pour but d'obtenir des modèles comparables deux à deux (cf. Fig 4.1) (la méthode de comparaison dépend de la méthode de classification utilisée).

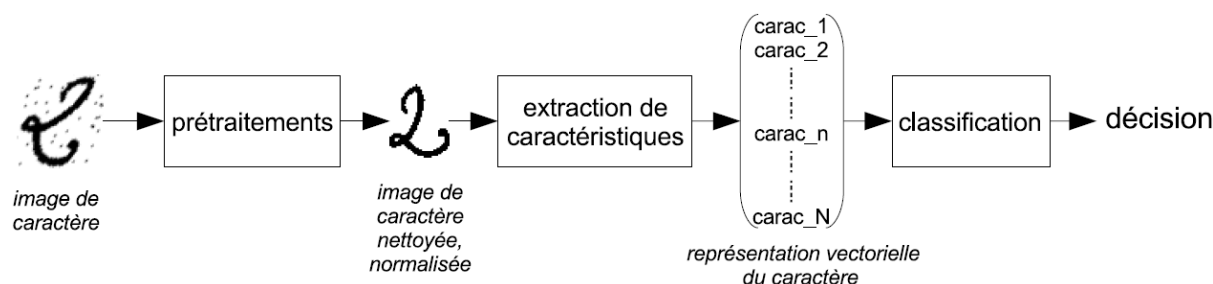


FIG. 4.1 – Tâches effectuées par l'OCR lors de la reconnaissance.

La solution intuitive pour comparer deux caractères est de comparer leurs images pixel à pixel. Dans la pratique, le problème est bien plus complexe que cela puisque, dans le cas de caractères manuscrits, il est très rare qu'une même personne écrive deux fois le même caractère de la même façon. C'est pourquoi des méthodes de caractérisation de ces caractères ont été développées. Dans la littérature, ces méthodes de caractérisation sont communément appelées soit simplement “caractéristique”, soit “signature”. La nuance

entre ces deux expressions est qu’une signature contient une ou plusieurs caractéristiques. Comme une caractéristique seule est rarement discriminante, nous en associons plusieurs afin d’obtenir une signature la plus robuste possible.

Un exemple de caractéristique simple est le nombre d’occlusions d’un caractère. Le “L” a zéro occlusion alors que le “8” en a deux, ou encore le ratio entre le nombre de pixels blancs et le nombre de pixels noirs. Nous pouvons citer aussi la méthode des sondes (cf. Fig. 4.2a) ou encore celle des profils (cf. Fig. 4.2b).

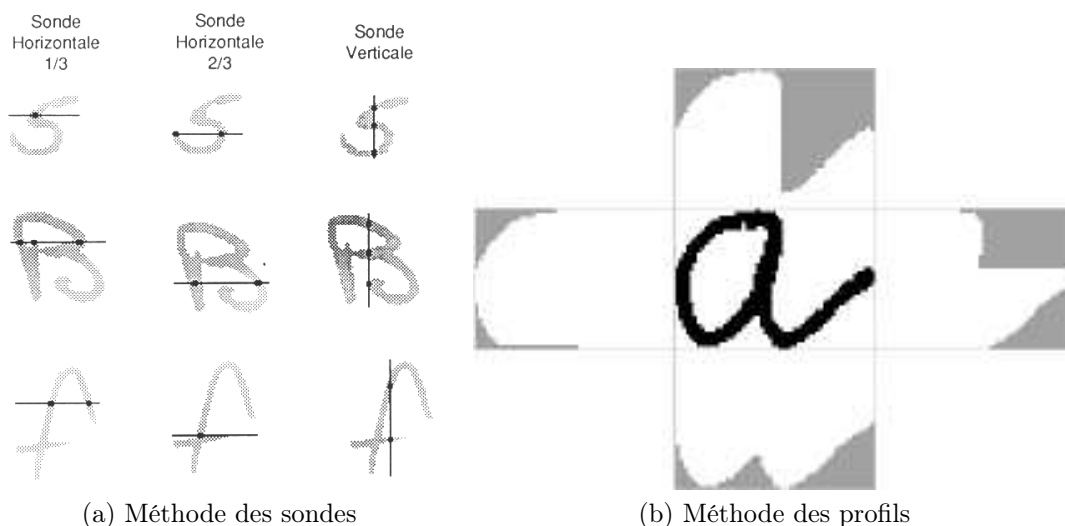


FIG. 4.2 – Méthodes statistiques de caractérisation de caractères.

Un exemple de signature est celle de Zernike ou plus précisément basée sur le calcul des “moments de Zernike” [16] [18]. Le calcul de ces moments est basé sur les polynômes découverts par Frederik Zernike, physicien néerlandais. Ces polynômes, très utilisés en optométrie, mettent en évidence les problèmes de déformations dues à l’utilisation de lentilles non parfaites (cf. Annexe 4). Nous pouvons parler de signature de Zernike car il ne s’agit pas d’une valeur numérique isolée. Nous pouvons citer aussi la transformée de Radon [15] [5] ou les moments de Hu [4].

4.1.2 Fonctionnement de notre OCR

Notre OCR étant basé sur des images venant de tracés vectoriels, nous n’avons pas besoin de l’étape de prétraitement ou plus précisément, cette étape est réduite à la simple transformation du tracé vectoriel en une image pixellisée. Cependant, la particularité de ce projet est de combiner deux types de valeurs caractéristiques. En effet, certaines de ces valeurs se calculent sur une image formée de pixels et d’autres peuvent être calculées sur le tracé temporel du caractère. L’idée est donc de combiner ces deux types de valeurs, “hors-ligne” et “en-ligne”.

4.2 Signatures

4.2.1 Contexte

La partie du projet concernant le calcul des caractéristiques a été basée sur le travail effectué par un ancien stagiaire. De ce fait, certaines portions de son code ont été entièrement reprises à partir d'autres travaux. Parmi ceux-ci, il y avait notamment les signatures basées sur les moments invariants de Zernike et la transformée de Radon. Par le passé, j'ai été amené à travailler avec ces signatures et à les recoder en Java pour le projet NaviGaLa (logiciel de classification et reconnaissance de symboles bruités utilisant les treillis de Galois comme classifieur [3]). Durant ce travail, avec mon collègue nous avons trouvé des bugs et des non-optimisations dans les calculs de ces signatures. Par ailleurs, nous avons aussi travaillé avec les transformées de Fourier-Mellin qui n'étaient pas incluses dans le package de calcul de caractéristiques. Après analyse du contexte, il fut décidé que ma première tâche serait de re-coder ces signatures, alors en Java, en C++.

4.2.2 Travail effectué

Pour effectuer ce travail, j'ai dû m'adapter aux bibliothèques disponibles pour les calculs mathématiques présents dans le projet, lesquelles n'étaient, bien entendu, pas les mêmes que celles que nous avons utilisées en Java. Lors de la réimplémentation des signatures, j'ai essayé le plus possible de conserver la structure d'héritage que nous avons mise en place en Java. De cette façon, le code ainsi obtenu était extensible à d'autres types de signatures pour le futur et restait suffisamment indépendant du projet pour pouvoir être réutilisé dans un autre projet sans modifications. En effet, dans ce travail, j'ai recodé la signature exploitant la transformée de Fourier-Mellin qui n'est pas utilisée dans ce projet. Ceci a été fait dans le but de rassembler en une même entité les différentes signatures implémentées dans le laboratoire. À la fin de ce travail, j'avais donc une classe abstraite et une classe pour chacune des trois signatures.

4.3 Les Caractéristiques

4.3.1 Contexte

Le démonstrateur ayant montré ses capacités lors de la présentation faite devant M. Clavaud, il fut décidé de mettre en pause son développement et d'accentuer le travail sur l'amélioration des résultats de la partie OCR. Pour cela, nous nous sommes répartis les tâches à effectuer et j'ai eu comme mission d'améliorer la gestion du calcul des caractéristiques hors-ligne. Ce travail a consisté à regrouper les caractéristiques ayant des relations entre elles dans des classes séparées. En effet, toutes ces caractéristiques étaient alors calculées dans une seule et unique classe. Ceci était gênant car le vecteur ainsi créé contenait alors 244 valeurs (incluant 32 de Zernike et 8 de Radon par exemple). Comme pour alléger les temps de calculs nous effectuons une sélection d'une trentaine de variables discriminantes sur les 244 (en moyenne), il paraissait illogique de toutes les recalculer de façon systématique.

4.3.2 Travail effectué

Après analyse, nous étions capables de “découper” ce vecteur de 244 valeurs en 12 classes différentes (cf. Annexe 5), ce qui augmenta alors de façon conséquente le nombre de classes présentes dans le package “Signatures” créé au début de mon stage. Ceci a comme résultat que si aucune des 32 caractéristiques de Zernike n’est sélectionnée, alors on n’effectue pas le calcul sur cette classe. Ce faisant, nous économisons du temps de calcul lors de la reconnaissance de caractère puisque pour chaque caractère à reconnaître, il faut calculer les caractéristiques de celui-ci.

Les caractéristiques hors-lignes ont donc pu être découpées selon la configuration suivante :

- Les intersections, embranchements et fin de traits
- Les extremas locaux
- Les taux de remplissage
- Les moments de Hu
- Les sondes horizontales, verticales et diagonales ainsi que les distances les séparant
- Les profils
- Les projections sur les axes
- La transformée de Radon
- Les moments invariants basés sur les polynômes de Zernike
- Des caractéristiques diverses isolées (nombre d’occlusions, élongation, compacité ...)

4.4 Analyse des résultats

En terme de gain de temps, nous ne pouvons donner aucune estimation étant donné que cela dépend complètement de la sélection des caractéristiques effectuée lors de l’apprentissage. En effet, si lors de la sélection, celle-ci donne comme réponse une liste de caractéristiques se retrouvant dans toutes les classes, alors il n’y aura aucun gain. Par contre, si lors de cette sélection aucune des caractéristiques de Radon ou de Zernike n’est conservée, alors le gain sera important étant donné le nombre et la complexité de calcul de celles-ci. Cependant, nous y avons gagné en terme de lisibilité, de modularité et d’efficacité pour la traque d’erreurs et/ou de fuite de mémoire.

Chapitre 5

Outils & contributions

5.1 Outils employés

Durant ce stage, j'ai été amené à manipuler différents outils en fonction des tâches accomplies.

Pour le développement du démonstrateur, j'ai principalement utilisé l'environnement de développement NetBeans 6.1 [12]. Cet environnement a comme avantages d'être libre d'utilisation, d'exploiter pleinement les technologies Java puisque lui-même codé dans ce langage et d'avoir un système très évolué de génération d'interface graphique. Je n'ai que très peu utilisé ce dernier car je souhaitais avoir un code le plus léger et le plus structuré possible afin de pouvoir rajouter facilement des fonctionnalités durant le développement. En effet, ce système produit un code de bonne qualité mais très difficilement modifiable a posteriori. C'est pourquoi j'ai limité son utilisation à la création rapide d'ébauches d'interfaces lors de la conception. Par la suite, j'ai utilisé les outils de base fournis par le Java Development Kit (JDK). La compilation a été effectuée en utilisant la version 6 du JDK [7] mais avec comme cible une compatibilité avec la JVM version 5 [6].

Pour la manipulation du format SVG, du côté programmation, j'ai utilisé la bibliothèque "open source" Batik 1.7 [2] qui est développée par l'Apache Software Foundation et qui se trouve être écrite en Java. Par ailleurs, il s'agit aussi de la bibliothèque la plus complète concernant la gestion du langage SVG. Pour m'aider à comprendre les différentes subtilités de ce langage, j'ai utilisé le logiciel Safari [1], navigateur internet, qui a la particularité d'être l'outil de visualisation de document SVG le plus respectueux de ce langage.

Pour la programmation en C++ des signatures et des caractéristiques, j'ai utilisé le logiciel Visual Studio Express 2008 [10] qui a comme avantage d'être gratuit et pleinement compatible avec la version du compilateur utilisée par nos partenaires belges. Pour ce qui concerne la partie traitement d'image, j'ai utilisé la bibliothèque OpenCV [13] qui était déjà employée par mes collègues pour d'autres aspects du projet et pour les calculs numériques, la bibliothèque standard du langage C++ a suffi.

Pour la rédaction de ce rapport, j'ai utilisé, sous Windows, TeXnicCenter [17] et, sous MacOS X, TextMate [8] qui sont des éditeurs de texte ayant des fonctionnalités de coloration syntaxique reconnaissant la syntaxe du langage $\text{\LaTeX}$. Pour la compilation de ce rapport, j'ai utilisé, sous Windows, la distribution MikTeX 2.7 [11] et, sous MacOS X, la distribution MacTeX 2007 [9].

5.2 Contributions

Tout au long de ce stage, j'ai aussi apporté diverses contributions autour de moi d'un point de vue recherche comme d'un point de vue développement. Ces contributions ont, pour la plupart, été ponctuelles mais se sont souvent révélées utiles.

Du point de vue recherche, j'ai notamment aidé à la recherche d'informations sur les calculs des moments de Zernike afin de comprendre le fonctionnement du code que nous avons. Ceci avait pour finalité une optimisation de ce code de façon à éliminer des erreurs de calculs qui faisaient que nos signatures étaient erronées. En effet, ces calculs s'effectuent de façon incrémentale et dans l'ancienne version de ce code, ils étaient effectués séquentiellement ce qui nous donnait, d'une part, un code long et, d'une autre part, un code fastidieux à relire, à comprendre et potentiellement buggué. J'ai aussi participé à la recherche d'articles scientifiques mettant au point des méthodes de classification basées sur des machines à vecteur de support permettant de fournir en résultat des probabilités d'appartenance aux classes présentes dans la base d'apprentissage.

Du point de vue développement, j'ai été sollicité sur plusieurs petites choses. La première a été l'élaboration des formats de fichiers XML que nous avons établis pour stocker les modèles de formulaires comme les résultats du système de reconnaissance. Ensuite, j'ai apporté mes connaissances en bases de données pour aider Clément à mettre en place du Système de Gestion de Base de Données (SGBD) pour stocker les caractéristiques de la phase d'apprentissage. Notamment, je lui ai indiqué l'existence du SGBD SQLite3 [14] qui présente les avantages d'être "server-less" et d'être écrit en langage C. De ce fait, celui-ci correspondait parfaitement à nos besoins. Par ailleurs, mes connaissances de NaviGaLa ont été demandées afin de cerner le travail à faire d'un stagiaire crétois. En effet, ayant été co-développeur de ce logiciel lors de mon projet de Master 1, j'en connais assez bien les principaux défauts et, avec mon expérience, les méthodes employables pour essayer de les résoudre.

Conclusion

Ce stage fut très bénéfique pour moi car il m'a permis d'appréhender les impératifs industriels et commerciaux de très près tout en étant pas directement impliqué. De plus, il m'a permis aussi d'avoir la charge du développement d'un logiciel qui sera utilisé à l'avenir pour diverses occasions. Comme à l'avenir, j'aimerai travailler en tant qu'ingénieur d'études en développement d'application dans des laboratoires, ce stage m'a aussi donné un avant-goût de ce métier qui m'attire. De plus, durant celui-ci, j'ai élargi mes connaissances dans le domaine des classifieurs de données et de caractérisation d'image.

Bibliographie

- [1] Apple Safari 3.1
 . Apple Inc.
 . <http://www.apple.com/fr/safari/>.
- [2] Batik SVG Toolkit
 . Apache Software Foundation
 . <http://xmlgraphics.apache.org/batik/overview>.
- [3] Bertet K., Guillas S., Ogier J.-M. Extensions of Bordat's algorithm for attributes. In *Fifth International Conference on Concept Lattices and their Applications (CLA '2007)*, pages 38–49, Montpellier, France, october 2007. LIRMM and University of Montpellier II.
- [4] Hu M.K. Visual pattern recognition by moment invariants. *IRE Trans. on Information Theory*, 8 :179–187, 1962.
- [5] J. Radon. Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. *Berichte Saechsische Akademie der Wissenschaften*, pages 262–277, 1917.
- [6] Java 2 Platform Standard Edition (J2SE) 5.0
 . Sun Microsystems Inc.
 . <http://java.sun.com/j2se/1.5.0/docs/relnotes/features.html>.
- [7] Java 2 Platform Standard Edition version 6
 . Sun Microsystems Inc.
 . <http://java.sun.com/javase/6/features.jsp>.
- [8] MacroMates TextMate 1.5.7
 . MacroMates
 . <http://www.macromates.com/>.
- [9] MacTeX-2007
 . TeX User Groups
 . <http://www.tug.org/mactex/>.
- [10] Microsoft Visual Studio Express 2008
 . Microsoft Corporation
 . <http://msdn.microsoft.com/fr-fr/express/default.aspx>.
- [11] MiKTeX 2.7
 . MiKTeX Project
 . <http://miktex.org/>.

- [12] NetBeans IDE 6.1
 . Sun Microsystems Inc.
 . [http ://www.netbeans.org](http://www.netbeans.org).
- [13] OpenCV Library
 . [http ://opencvlibrary.sourceforge.net/](http://opencvlibrary.sourceforge.net/).
- [14] SQLite
 . [http ://www.sqlite.org/features.html](http://www.sqlite.org/features.html).
- [15] Tabbone, S. and Wendling, L. Recherche d'images par le contenu à l'aide de la transformée de Radon. *Technique et Science Informatiques*, 2003.
- [16] Teague, M. Image analysis via the general theory of moments. *Journal of Optical Society of America (JOSA)*, 70 :920–930, 2003.
- [17] TeXnicCenter 1 Beta 7.01
 . ToolsCenter.org
 . [http ://www.toolscenter.org/](http://www.toolscenter.org/).
- [18] Yziquel J. and Ermont N. Apprentissage par réseaux de neurones. Technical report, École Nationale Supérieure d'Ingénieurs de Caen (ENSICAEN), 2005.

Annexes

Annexe 5.1	Brochure DocLedge (anciennement e-Connector).
Annexe 5.2	Capture d'écran, première ébauche.
Annexe 5.3	Capture d'écran, version avancée.
Annexe 5.4	Modélisation des déformations selon les polynômes de Zernike.
Annexe 5.5	Arborescence des classes du Démonstrateur.
Annexe 5.6	Diagramme de classes global du Démonstrateur.
Annexe 5.7	Diagrammes de classes détaillés.
Annexe 5.8	Diagrammes de classes détaillés.
Annexe 5.9	Diagrammes de classes détaillés.
Annexe 5.10	Diagrammes de classes détaillés.
Annexe 5.11	Diagramme de classes des Signatures développées.

Présentation de l'entreprise CONNECTOR

Créée en 2002, la société CONNECTOR est aujourd'hui spécialisée dans la capture et la conversion de données informatiques.

Nos activités comprennent la reconnaissance de données manuscrites, la numérisation de documents papier, la reconnaissance du contenu, le traitement (indexation) du document électronique et de son contenu, l'archivage actif.



Pour répondre à ces attentes, CONNECTOR propose un portefeuille d'outils capables de traiter cette diversité quelle que soit la dimension de l'entreprise :

- Acquisition de données manuscrites avec RecogInk ;
- Acquisition automatique de données structurées ou non avec eConnector;
- Conversion de tout type de données informatiques avec XML2XML ;
- Capture et archivage automatique des e-mails avec @rcheMail ;
- Gestion des documents d'entreprise avec @rcheDoc ;
- Enterprise library management : maîtrise du cycle de vie des documents et dossiers électroniques;
- Full text retrieval : recherche de l'information souhaitée.

Les études des bureaux de consultance nous font constater que la quantité de documents papiers à archiver ne cesse de croître, la recherche des documents est de plus en plus fastidieuse, la diffusion de l'information est mauvaise, le nombre de sites internet augmente mais leur contenu n'est pas régulièrement actualisé,...

Notre mission est d'augmenter les avantages compétitifs de nos clients en matière de gestion de l'information.

Les bénéfices de nos solutions ne sont pas négligeables : diminution du coût de stockage et de traitement des documents, sécurité accrue grâce aux backups et gain de temps majeur pour le personnel qui peut enfin se concentrer sur son métier.

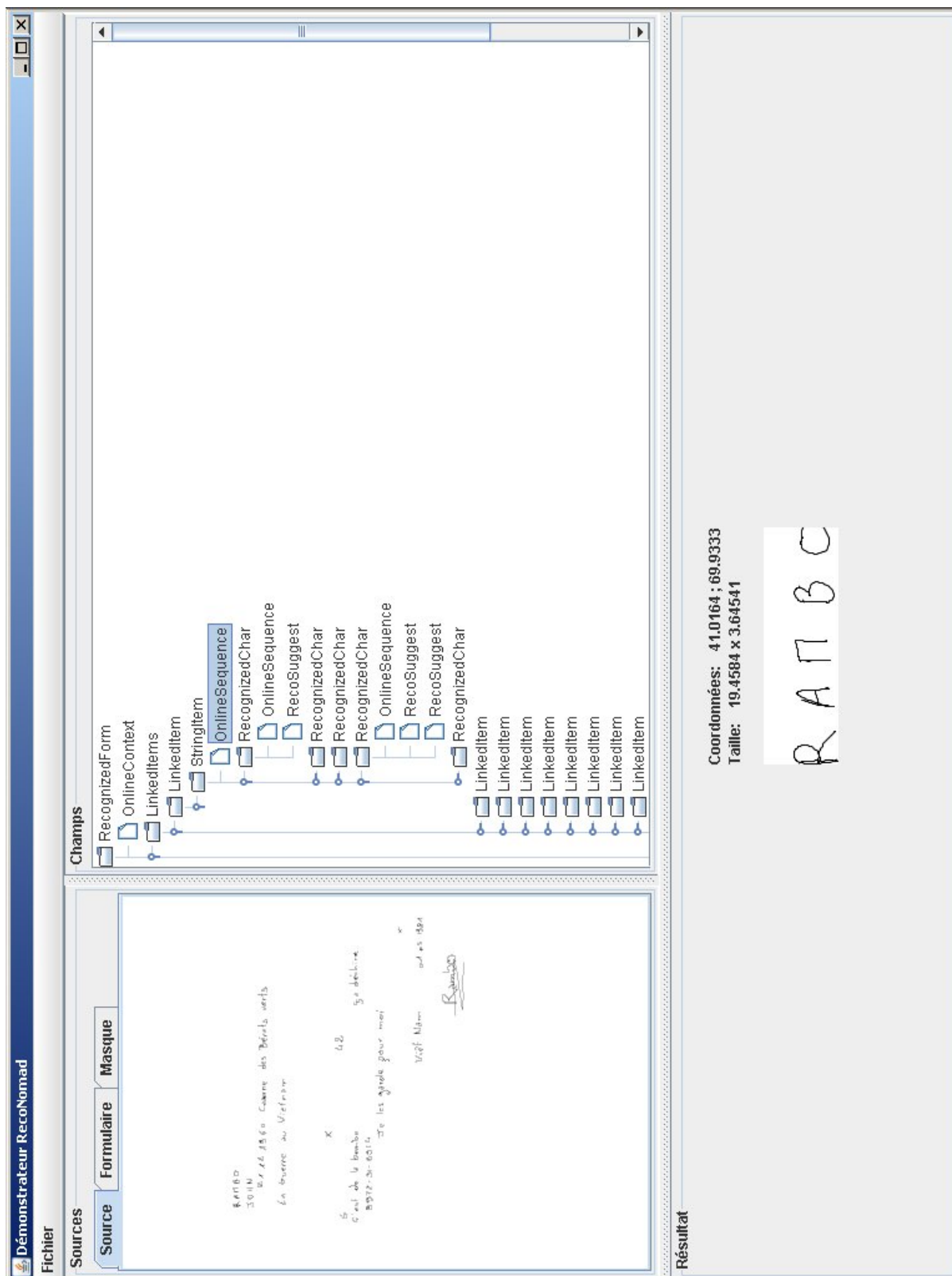
CONNECTOR G.I.E.

Rue Saint-Vincent, 35c

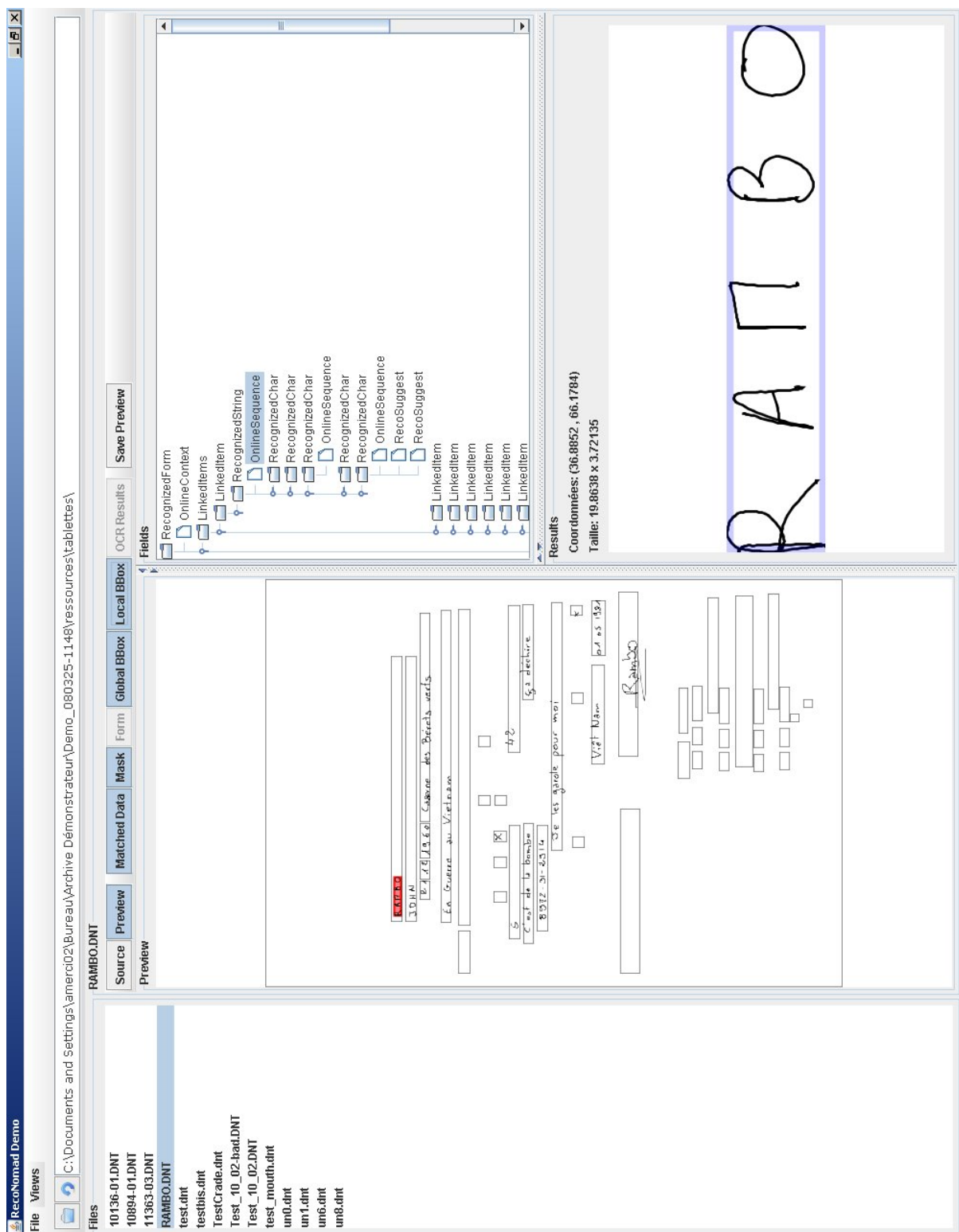
B-1457 Nil-Saint-Vincent

T. +32 10 654996 - F. +32 10 654991

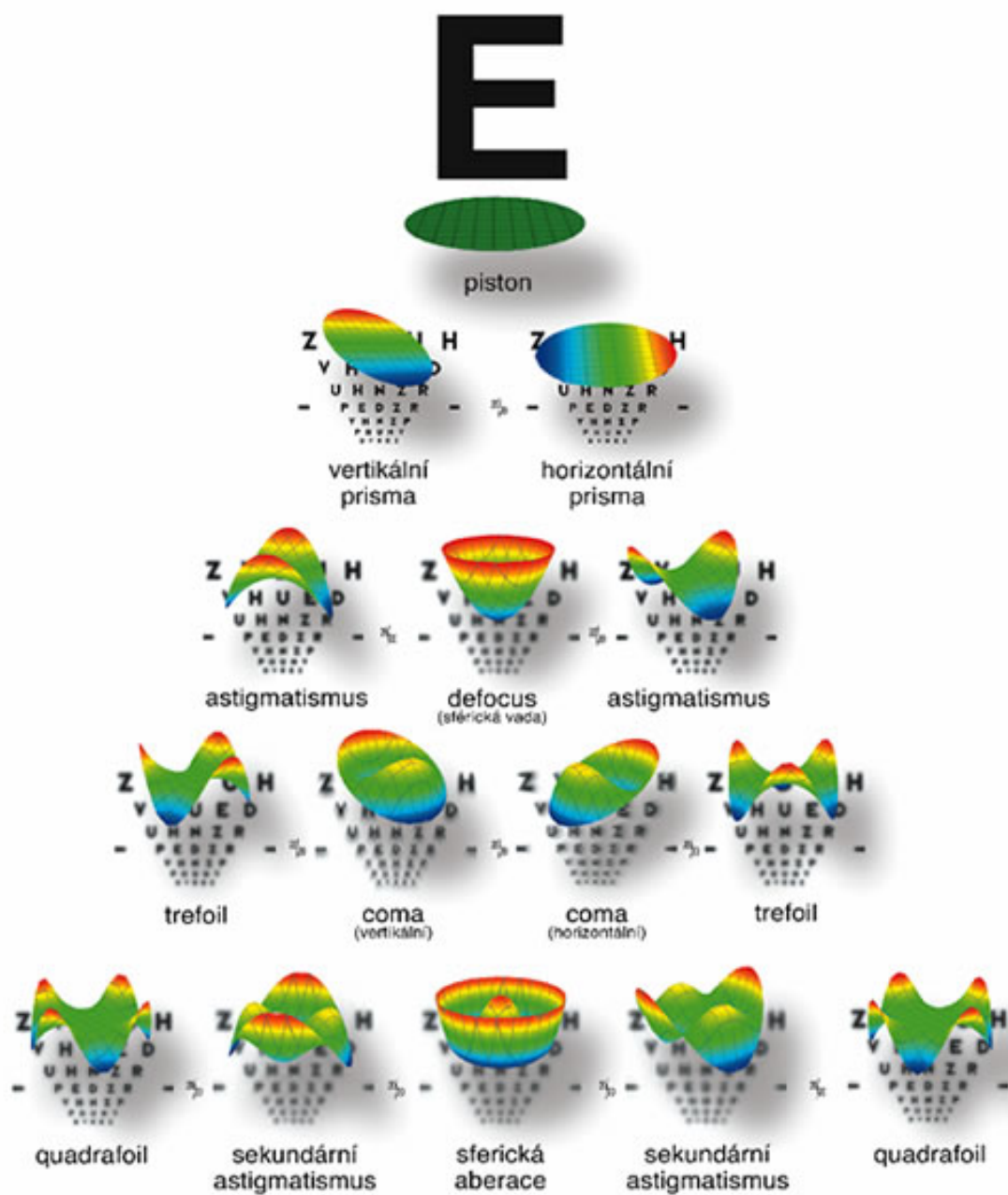
info@econnector.biz



Annexe 5.2 – Capture d'écran, première ébauche.

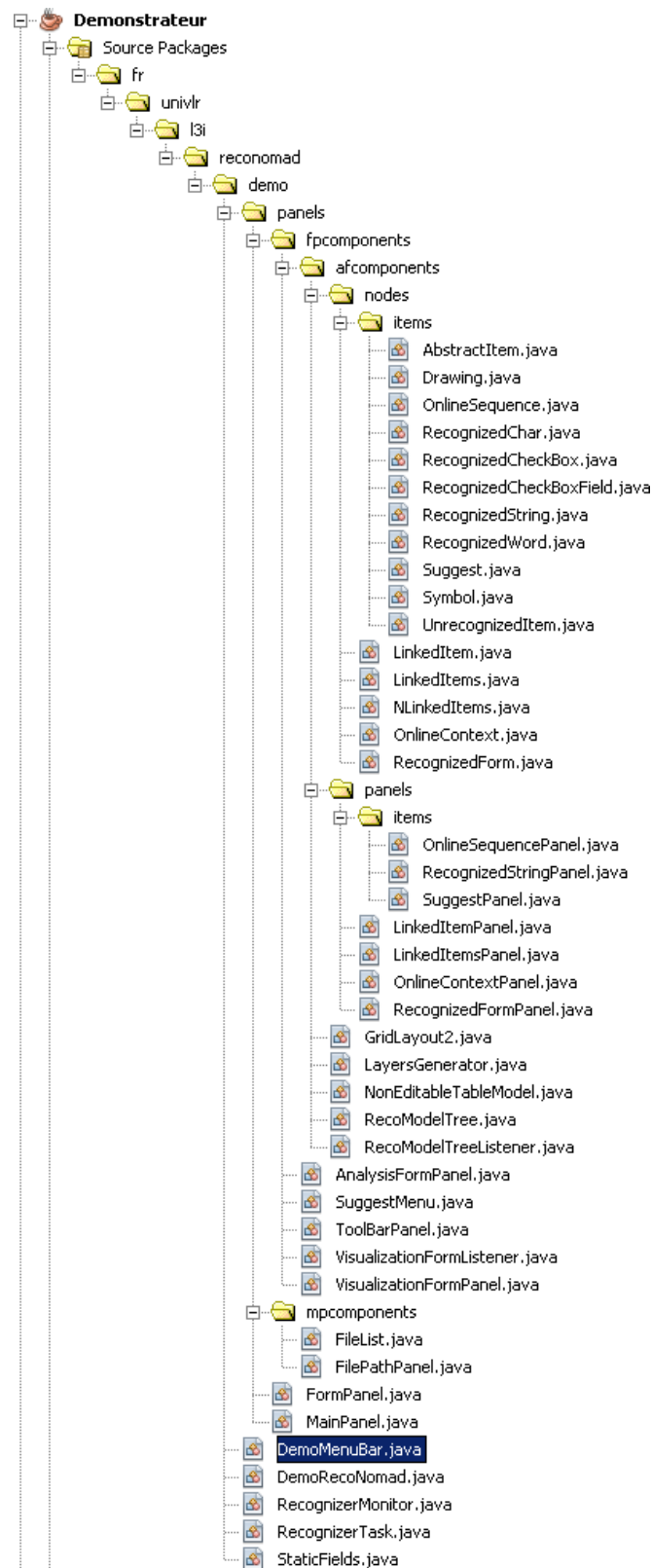


Annexe 5.3 – Capture d'écran, version avancée.

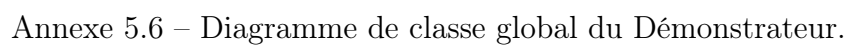


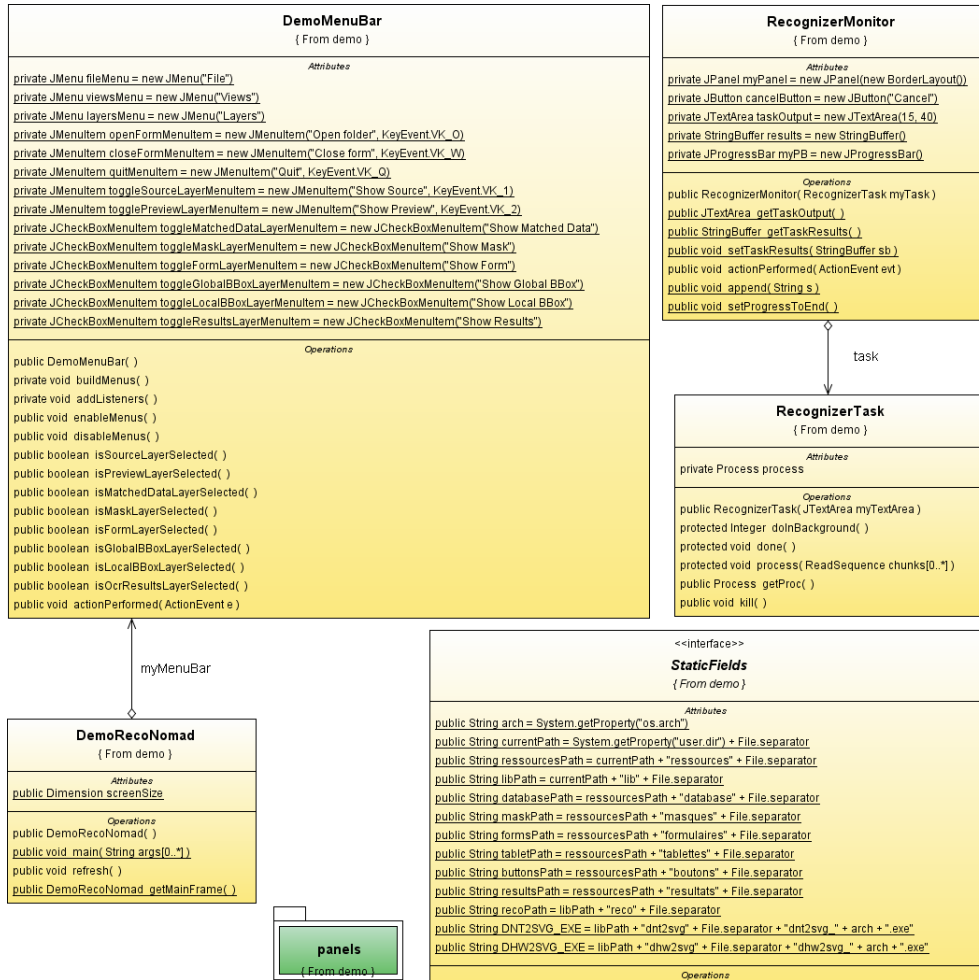
LADARVISION
CUSTOMCORNEA

Annexe 5.4 – Modélisation des déformations selon les polynômes de Zernike.

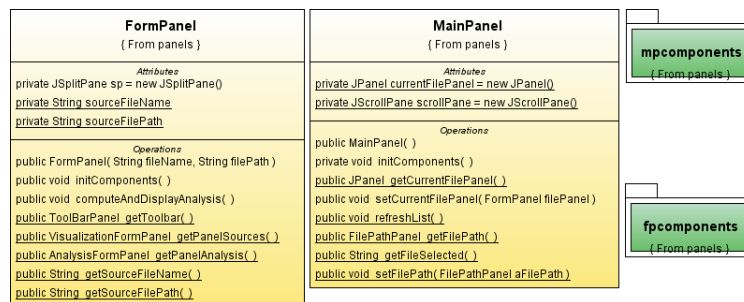


Annexe 5.5 – Arborescence des classes du Démonstrateur.



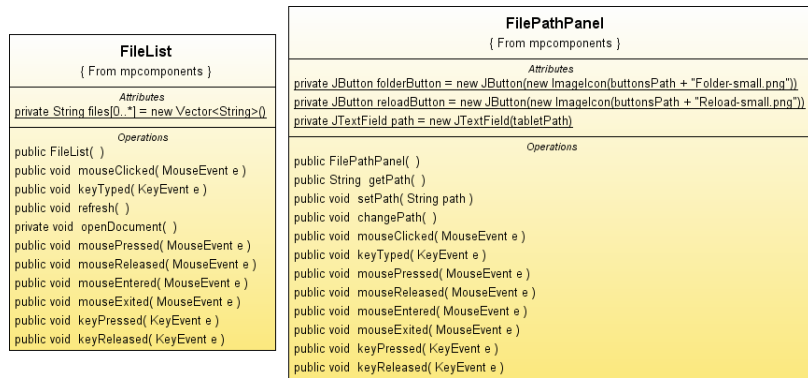


(a) Diagramme de classe du package “demo”.

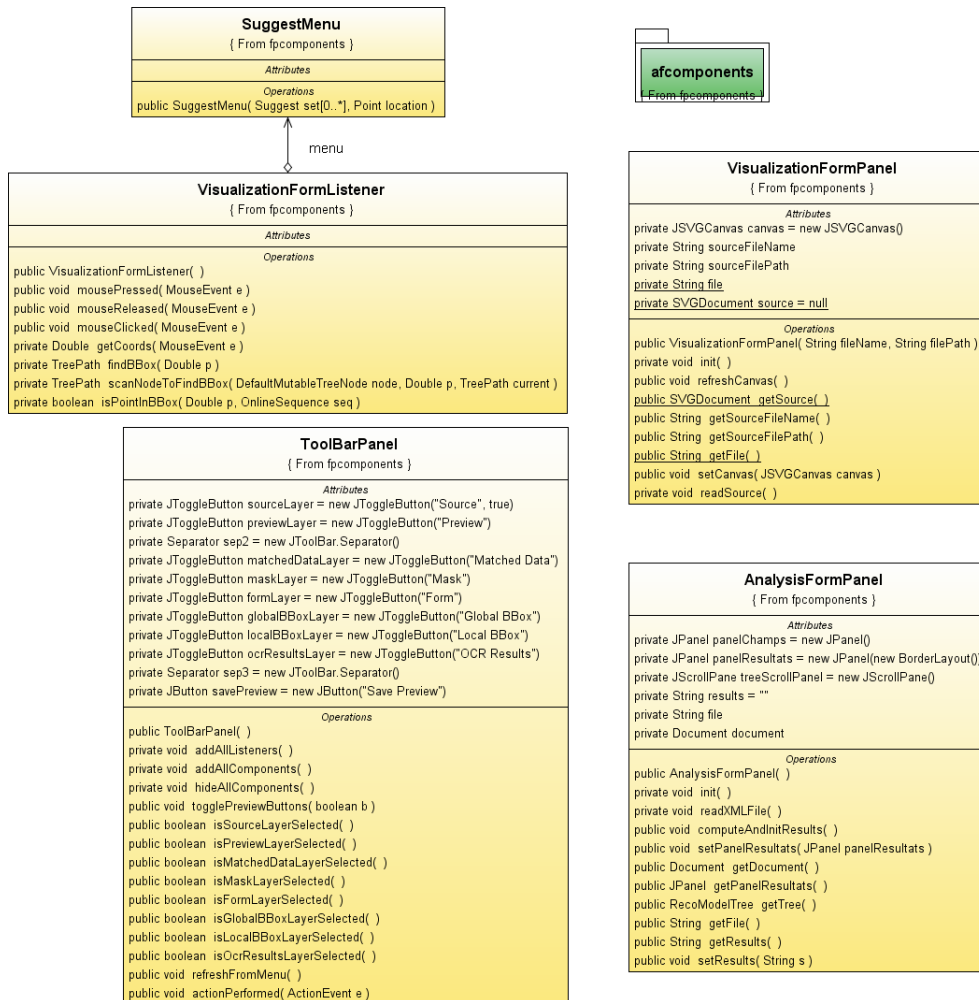


(b) Diagramme de classe du package “panels”.

Annexe 5.7 – Diagrammes de classes détaillés.

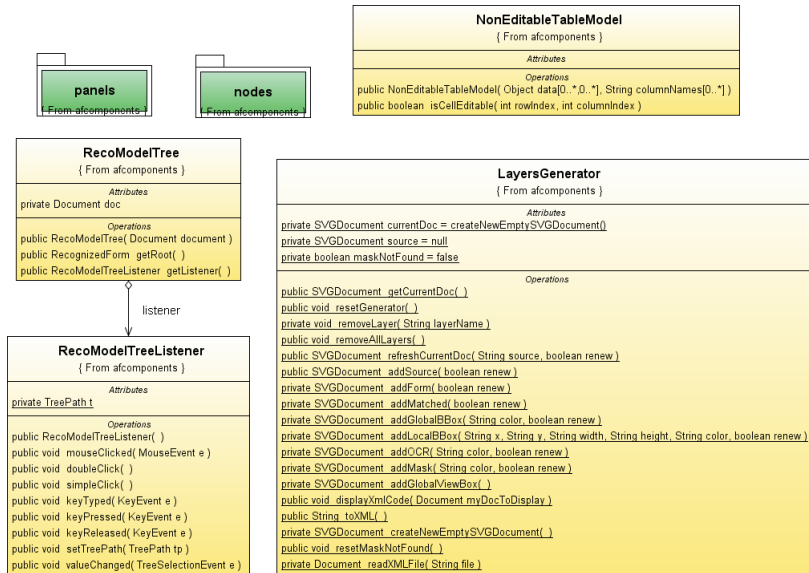


(a) Diagramme de classe du package “mpcomponents”.

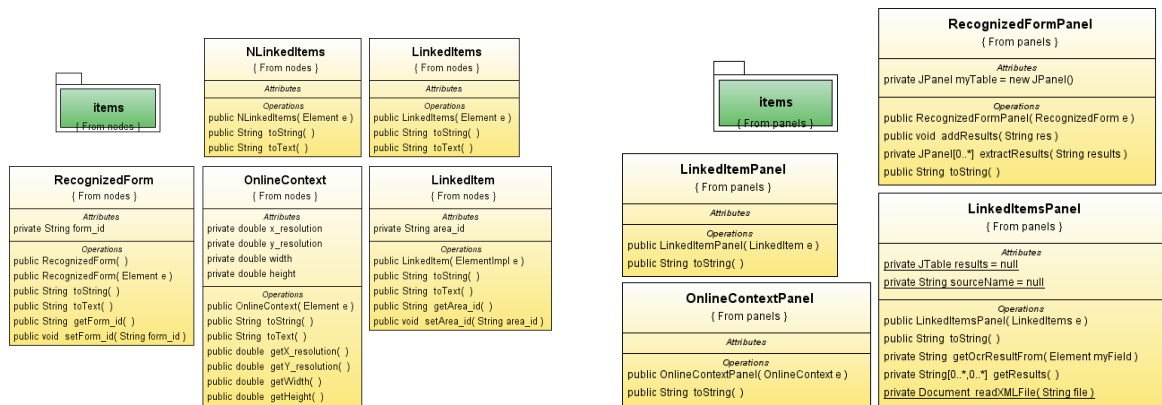


(b) Diagramme de classe du package “fpcomponents”.

Annexe 5.8 – Diagrammes de classes détaillés.



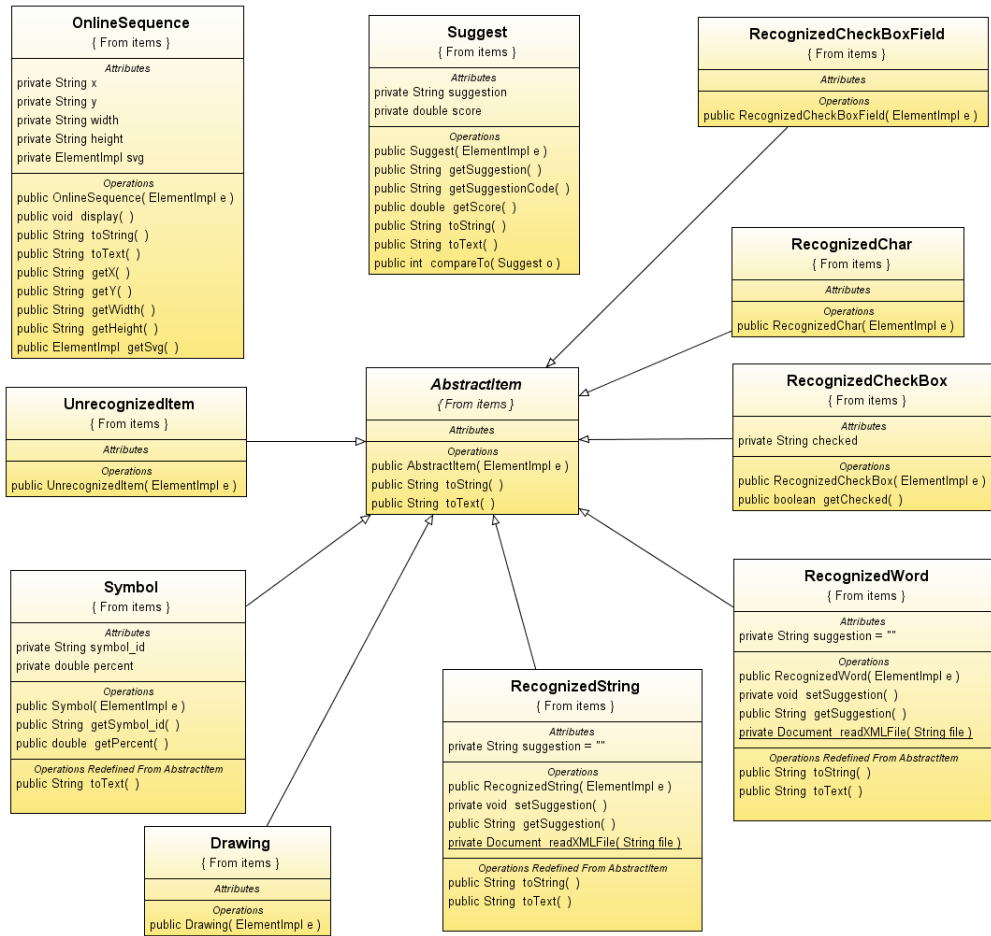
(a) Diagramme de classe du package “afcomponents”.



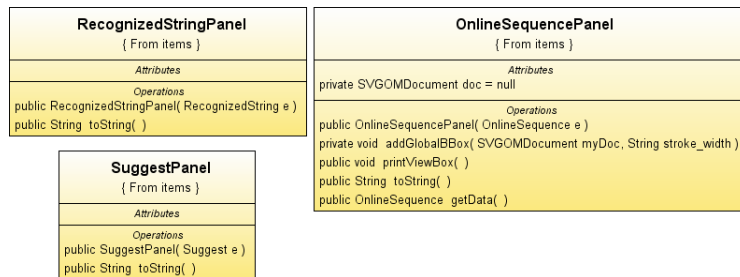
(b) Diagramme de classe du package “nodes”.

(c) Diagramme de classe du package “panels”.

Annexe 5.9 – Diagrammes de classes détaillés.

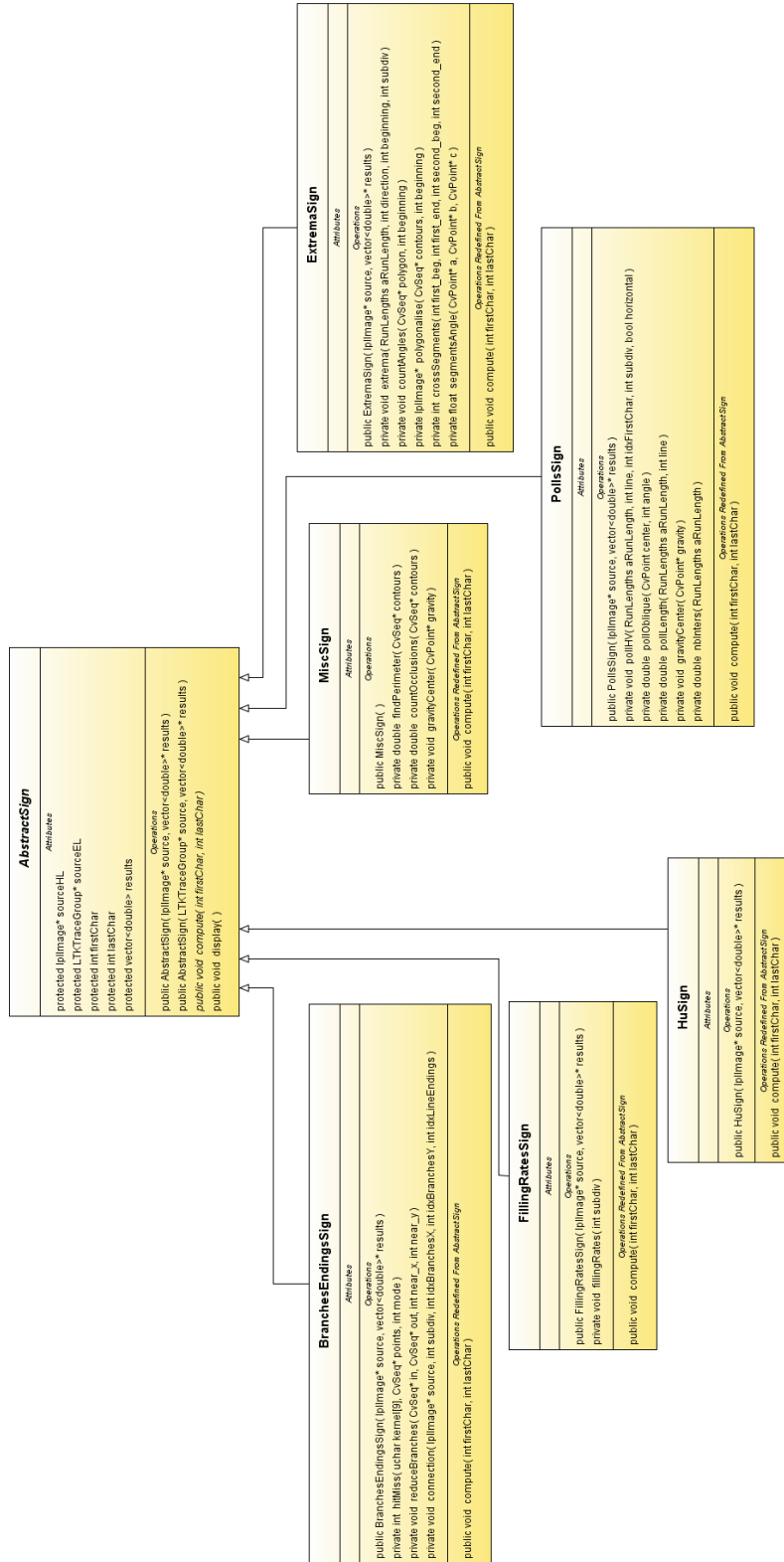


(a) Diagramme de classe du package “nodes.items”.



(b) Diagramme de classe du package “panels.items”.

Annexe 5.10 – Diagrammes de classes détaillés.



Annexe 5.11 – Extrait du diagramme de classes des Signatures implémentées.