

L'intégration des données dans un système de reconnaissance de documents assistée

Frédéric Bapst Rolf Brugger Abdelwahab Zramdini Rolf Ingold

IIUF, Université de Fribourg, Chemin du Musée 3, CH-1700 Fribourg

frederic.bapst@unifr.ch

Résumé

Dans le cadre d'un nouveau projet en reconnaissance de documents assistée, nous nous attaquons au problème de l'intégration harmonieuse des données à gérer pendant une session de travail. En partant des besoins suscités par le projet *CIDRE*¹, nous analysons les informations à traiter pour en proposer une modélisation solide. Nous aboutissons à la définition d'un format pour représenter une session de reconnaissance, basé sur la librairie DAFS.

Mots-clé : reconnaissance de documents structurés, intégration, format de document, environnement interactif.

1 Introduction

Le projet *CIDRE* [3] (pour Cooperative & Interactive Document Reverse Engineering) s'inscrit dans une lignée de travaux menés par notre groupe de recherche depuis près de six ans dans le domaine de l'analyse de documents. Il est fondé sur une ré-évaluation générale de la problématique en reconnaissance de documents structurés. Dans la conception d'une architecture logicielle pertinente, le paradigme de l'*intégration* s'impose comme thème directeur avec de multiples facettes dignes d'intérêt, comme l'illustre la figure 1. Le présent article se focalise sur l'intégration des données.

Les systèmes développés à ce jour ont montré qu'une reconnaissance automatique, conçue comme une boîte noire qui consomme des images et produit la structure logique en appelant des traitements selon un schéma séquentiel simple, est illusoire. Nous sommes convaincus que l'intervention par un opérateur humain est essentielle, et que cet aspect coopératif doit se manifester

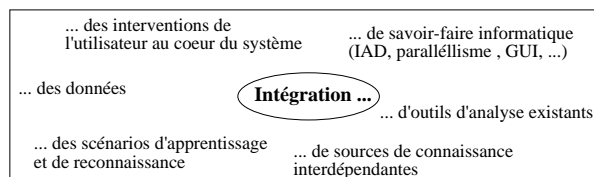


Figure 1: Formes d'intégration dans le projet *CIDRE*.

tout au long de l'évolution d'une session de reconnaissance. L'intégration des données est une première étape vers la conception d'une architecture logicielle réellement collaborative.

Le deuxième chapitre résume les fondements de notre approche concernant la convivialité, l'apprentissage de modèles, l'architecture logicielle et les applications potentielles, autant d'aspects qui positionnent clairement le projet par rapport à ce qui a déjà été fait. La notion d'intégration des données est approfondie au chapitre 3, qui met d'abord l'accent sur les motivations de cette étape de conception, puis dresse un cahier des charges en établissant l'inventaire des données à gérer. Le chapitre 4 présente notre solution, qui consiste en une modélisation des informations adaptée au format DAFS.

2 Ambitions du projet *CIDRE*

La figure 2 situe le rôle de l'environnement *CIDRE* dans la phase de re-structuration de documents. On y décèle plusieurs des ambitions qui font tout l'intérêt du projet.

2.1 Convivialité

Le point de départ du projet consiste en une révision des fonctionnalités du système à

¹ce projet est soutenu par le Fonds National suisse pour la Recherche Scientifique, subside no 21-42'355.95

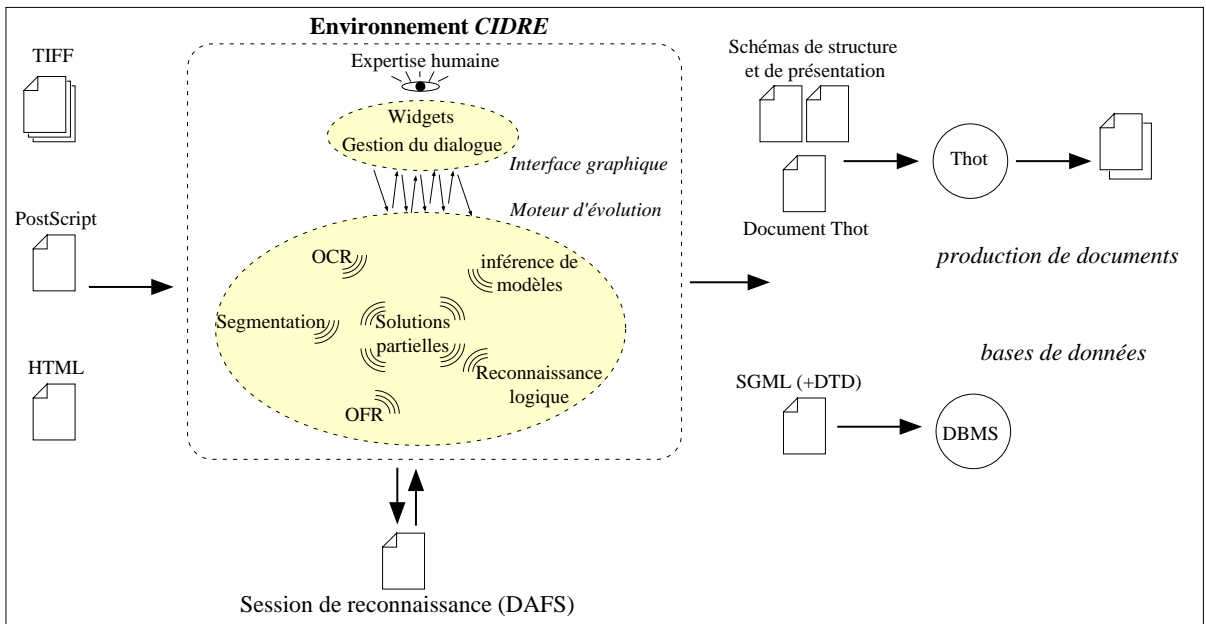


Figure 2: La ré-ingénierie du document avec *CIDRE*.

développer : nous abandonnons clairement l'objectif d'aboutir à un outil de reconnaissance 100% automatique, pour viser plutôt le développement d'un environnement d'aide à la restructuration de documents.

L'expérience a montré qu'il est illusoire de produire un système de reconnaissance entièrement automatique qui soit opérationnel, à moins de cibler une unique application très restreinte. Au lieu de bricoler des fonctions de corrections manuelles autour d'un outil qui les exclut par nature, c'est toute la conception du système qui doit être imprégnée du souci d'assister au mieux l'utilisateur. Fondamentalement, il faut revaloriser le rôle de l'opérateur humain pour le rendre maître de la tâche de reconnaissance [5]; les fonctions d'analyse automatique sont offertes pour qu'il puisse en profiter, non pour qu'il les subisse. A notre avis, la définition d'un environnement convivial n'est pas qu'une question banale d'interface graphique, car la prise en compte des interventions de l'utilisateur a une incidence sur l'architecture du système.

2.2 Inférence automatique de modèles

Ce côté hautement interactif provoque une révision des hypothèses sur les informations disponibles : alors que la grande majorité des travaux précédents [11] [7] supposaient l'exis-

tence préalable d'une description formelle des modèles de documents à reconnaître, notre système doit être capable de reconstituer de manière incrémentale le modèle durant la session de reconnaissance interactive [6] [1].

En effet, la génération d'une description détaillée demande un tel effort que rares sont les situations où cela en vaut la peine. Ce n'est certainement pas conciliable avec un système qui soit capable de s'adapter aux besoins quotidiens de l'utilisateur. De plus, une description formelle ne tient pas compte des imperfections, voire des inconsistances délibérées auxquelles on fait face en pratique. Ces remarques montrent la difficulté d'utiliser un système paramétré par un modèle formel, mais ne remettent pas en cause la notion de document générique : au contraire, celui-ci n'est plus considéré comme un fichier de configuration passif, mais plutôt comme une connaissance qui se fait enrichir lors de chaque reconnaissance; on pourrait même envisager de réutiliser ce modèle inféré pour *produire* de nouveaux documents conformes dans un éditeur structuré comme Thot [16].

2.3 Architecture coopérative

Les deux objectifs précédents nous ont amenés à réorienter les axes de recherche : plutôt que de continuer à approfondir isolément des sujets très pointus, nous préférons porter nos efforts sur une

intégration harmonieuse du savoir-faire actuel.

La quasi-totalité des traitements impliqués dans la reconnaissance de documents ne seront jamais totalement fiables. On pourra continuellement en améliorer les performances dans un contexte précis ou évaluer de nouveaux algorithmes, mais il ne faut pas attendre de progrès sensibles et directement profitables à l'utilisateur : plus on affine un sujet, plus on pose d'hypothèses, et plus on s'éloigne des conditions d'une utilisation pratique. Il nous semble que les défauts inhérents aux outils d'analyse peuvent être surmontés si on développe une architecture coopérative qui admet le phénomène de remise en cause, qui tire un profit maximum des interdépendances entre sources de connaissances, et qui est sous le contrôle de l'opérateur humain. En somme, nous voulons éviter une chaîne de traitements rigide pour définir un moteur d'évolution plus adaptatif.

Notre groupe dispose d'une panoplie d'outils de base pour les prétraitements d'images, la reconnaissance de structures physiques [2], l'OCR, la reconnaissance de fontes [18] ou la reconnaissance logique. Ils forment un équipement indispensable pour axer notre recherche sur une intégration intelligente de tout le savoir-faire impliqué. Les architectures à base de tableaux noirs ont déjà fait l'objet de recherches intensives en reconnaissance de documents [7] ; nous sommes plus attirés par une décentralisation du contrôle et des données, basée sur le paradigme multi-agents.

2.4 Applications

Finalement nous profitons d'étendre le champ des applications potentielles : la reconnaissance de pages scannées demeure un domaine privilégié, mais le projet s'ouvre aussi sur la re-structuration de documents électroniques.

En effet, la prolifération des documents électroniques introduit de nouveaux besoins en matière de transformation de documents : posséder une version électronique d'un document ne signifie pas que celui-ci se trouve sous une forme exploitable; même si un format comme SGML [10] régnait comme standard universel, la structure désirée d'un document dépend toujours de l'utilisation qu'on veut en faire. Concrètement, les documents sont le plus souvent échangés sous une forme peu structurée et utilisés sous une forme plus riche : PostScript (dans une moindre mesure RTF ou PDF) s'impose comme le standard pour

la transmission à travers le réseau. Les pages HTML prennent une importance phénoménale pour l'échange d'informations, mais malgré des contenus souvent analogues (p. ex. les pages de présentation personnelles), chaque page définit son propre modèle; on voit clairement l'intérêt de collectionner ces documents de manière uniforme. L'environnement *CIDRE* vise aussi ces deux applications, avec l'avantage de travailler sur des données initiales non bruitées.

3 Objectifs de l'intégration

3.1 Intérêt d'une représentation uniforme

On pourrait être tenté de considérer la représentation des données comme un détail d'implémentation. Mais dans le contexte du projet *CIDRE*, plusieurs arguments militent en faveur d'une modélisation préalable des informations présentes.

Les différents outils d'analyse automatique (OCR, segmentation, reconnaissance de fonte etc.) sont interdépendants. En effet, un résultat partiel n'est en général pas du ressort d'une tâche unique. Ainsi les coordonnées d'un caractère peuvent provenir d'un regroupement/éclatement de composantes connexes, mais elles sont aussi attestées par l'OCR (qui procède à une autre segmentation). De même, on peut disposer d'une reconnaissance de fonte, mais le taux d'erreur d'un bon OCR monofonte est aussi a posteriori une indication sur la confiance à accorder à la police choisie. On est ainsi amené à préciser le rôle de chaque donnée (et de chaque requête d'analyse) si l'on tient à ce que les résultats soient mis en commun de manière sensée, c.-à-d. en remédiant aux éventuelles incohérences. C'est le premier pas pour évaluer différentes formes de coopération et de compétition entre sources de connaissances.

Non seulement les outils d'analyse participent conjointement à la construction d'une solution, mais surtout l'incertitude qui enveloppe chaque résultat partiel obligera de mettre en jeu des stratégies sophistiquées pour explorer l'espace des solutions; ainsi la solution courante évolue bien sur par l'incorporation de nouveaux résultats, mais aussi par la remise en cause de composants existants (on touche là le coeur même du moteur d'évolution d'une reconnaissance de documents). Ces révisions locales successives ne seront performantes que si les données se trouvent sous une

forme adaptée et homogène.

La revalorisation du rôle de l'utilisateur passe par une présentation soignée qui prévoit des primitives de navigation, mais aussi de modification de la solution en cours. En fait, le pilotage global d'une session de reconnaissance nécessite un accès adéquat à tous ses constituants. Dans le même ordre d'idée, il sera très utile de pouvoir matérialiser le concept de session de reconnaissance sous la forme d'un unique fichier (ce qui commence à être le cas pour des logiciels d'OCR modernes comme TypeReader [9]). La solution la plus élégante consiste à représenter l'intégralité des informations dans un unique format interne qui puisse être sauvé et chargé.

Enfin, il s'agit de préparer le terrain pour différentes extensions, telles que l'utilisation de nouveaux outils d'analyse (p. ex. un vérificateur lexical), ou la traduction depuis/vers d'autres formats de documents. Là aussi on a tout à profiter d'une représentation uniforme de la session en cours.

3.2 Données à intégrer

Les différentes informations qui participent à la constitution d'une session de reconnaissance se regroupent naturellement en plusieurs catégories :

Document spécifique – Le groupe d'information le plus intuitif concerne l'instance du document qui est en train d'être reconnu. On a l'habitude de le définir par sa structure logique (qui reflète la sémantique perçue par le lecteur) et sa structure physique (qui résulte de l'opération de formattage qu'il a subi) [12]. Ces deux structures hiérarchiques évoluent en cours de session et forment ensemble la solution provisoire actuelle; il s'agit donc de gérer cette dualité, et toutes les informations qui sont rattachées à leur constituants : étiquettes logiques, attributs typographiques (p. ex. fonte) et caractéristiques physiques (p. ex. dimensions).

Modèle générique – Le document spécifique en cours est toujours traité comme une instance précise d'un certain type de document, conformément à une convention maintenant bien établie [15]. Sur ce point, l'originalité du projet *CIDRE* réside dans la faculté de reconstruire ce modèle de documents, par un phénomène d'effet de bord produit par la reconnaissance de ses exemplaires. Ainsi le modèle générique, dont on est en train de

récupérer une instance, évolue au cours de la session de manière orthogonale à la solution spécifique.

Exploration de l'espace des solutions – La reconnaissance de document ne peut pas être réduite à un algorithme déterministe qui trouve directement la bonne solution, surtout en raison de l'incertitude qui entache toutes les données (bruit), voire à cause de réelles ambiguïtés. Le système doit plutôt explorer l'espace des solutions possibles et proposer à l'utilisateur la plus probable. Ainsi, outre le document courant tel qu'il est présenté à l'opérateur, il faut prévoir la gestion d'alternatives locales qui seront peut-être choisies pour la solution finale. Cet aspect concerne le fondement du moteur de reconnaissance, et doit être pris en compte dans l'intégration des données.

Etat de l'interface homme-machine – Nous tenons à manifester le côté hautement interactif de *CIDRE* en ménageant dans le concept de session une place pour l'état du dialogue homme-machine; ceci regroupe des paramètres de visualisation des entités (p. ex. couches d'informations), la gestion des modes de fonctionnement, la détermination d'un élément courant (p. ex. page courante) etc..

Paramétrisation des outils d'analyse –

De même que l'utilisateur est plongé dans un certain contexte de travail, ainsi en va-t-il pour les outils d'analyse automatique qui peuvent selon les cas être configurés avec de nombreux paramètres : fichiers d'apprentissage pour un OCR, seuils pour la segmentation, modèles de fontes pour l'OFR, combinaisons de techniques de prétraitements d'images etc.. Ces paramètres contribuent à augmenter la qualité des résultats proposés, et peuvent être ajustés durant la session.

4 Mise en oeuvre de l'intégration

Maintenant que la problématique de l'intégration des données est un peu mieux spécifiée, nous allons décrire notre proposition concrète qui se base sur le format DAFS [17]. Nous esquisserons rapidement les spécificités de ce format, puis les principes de base d'une utilisation pour notre projet; suivent une description détaillée des différents

éléments qui définissent notre spécialisation de DAFS, et enfin des indications sur la manière dont on envisage l'évolution des données lors de l'exécution.

4.1 Survol du format DAFS

DAFS, développé par RAF Technology, se veut un support pour le ré-encodage d'images de documents, en définissant un type abstrait sous forme de librairie C et une spécification du format de fichier. Le paquetage complet² contient aussi une application appelée *lluminator*, qui est un éditeur interactif de documents DAFS. A notre avis DAFS semble destiné à devenir un standard en la matière conformément à l'un de ses objectifs, mais il existe d'autres formats comparables [13].

A la base, DAFS est tout d'abord un gestionnaire d'arbre, étant donné que c'est la structure hiérarchique qui est la mieux adaptée à la gestion des documents structurés. Une *entité DAFS* est en fait un noeud d'arbre, qui représente typiquement une unité du document à reconnaître, comme dans la partie gauche de la figure 5. Les auteurs du format ont étendu ce concept en distinguant les noeuds ET et les noeuds OU, dont les fils ne symbolisent pas la composition, mais des points de vues alternatifs sur l'entité. Cela permet entre autres de cumuler dans la même structure la découpe physique et la découpe logique, ou de gérer plusieurs ordres de lectures.

Cette gestion d'arbre est complétée par une gestion d'images et de texte. Chaque noeud peut faire référence à une *image* (par défaut celle du père) et y occuper une surface précise. De plus, chaque entité peut posséder un *contenu*, avec des solutions spécialement prévues pour traiter les contenus textuels (ASCII ou même UNICODE).

Finalement, le format est extensible puisqu'on peut définir des *propriétés*, identifiées par des chaînes de caractères, qui permettent d'associer toutes sortes de données aux entités, p. ex. la fonte. De plus, on a l'occasion de définir de nouveaux *types d'entités*. A titre d'exemple, certaines entités classiques (blocs, lignes etc.) sont prédéfinies.

²disponible sur le réseau depuis le serveur DIMUND à l'adresse <http://documents.cfar.umd.edu/>

4.2 Organisation d'une session CIDRE en DAFS

Les entités à gérer durant une session de reconnaissance se partitionnent selon les deux axes générique/spécifique puis logique/physique. Les principales relations à établir entre entités de ces quatre groupes, illustrées sur la partie gauche de la figure 4, se regroupent en 6 catégories :

1. Les relations composition/contenant dénotent la décomposition structurelle des entités spécifiques. Chaque instance logique peut être décomposée en éléments logiques (sa composition naturelle) ou physiques (sa composition duale). Le cas des instances physiques est analogue. La structure logique prévoit des types de base, à savoir document, partie, fragment, chaîne et caractère. Pour la structure physique, on admet les types volume, page, région, bloc, ligne, mot, signe et séparateur [2]. La figure 5 montre les relations établies entre une ligne et une chaîne concrète. Notons enfin que ces relations encodent indirectement les relations de voisinage.
2. A l'intérieur de la structure logique, on peut trouver des liens logiques sémantiques particuliers, tels qu'un renvoi à une note de bas de page ou une référence bibliographique.
3. Les relations classe/instances indiquent la conformité des entités spécifiques (physiques ou logiques) à un équivalent générique.
4. Les entités génériques s'organisent par spécialisation successive des types de base. Les classes logiques sont désignées par une étiquette sémantique choisie par l'utilisateur. Les catégories physiques sont formées automatiquement par classement des entités génériques, pour constituer des groupes d'entités partageant des caractéristiques (p. ex. géométriques ou typographiques) communes. Ces deux organisations, qui autorisent une gestion modulaire du modèle, sont illustrées sur la figure 3.

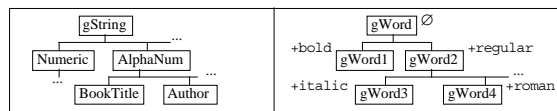


Figure 3: Spécialisation hiérarchique d'étiquettes logiques et de classes physiques.

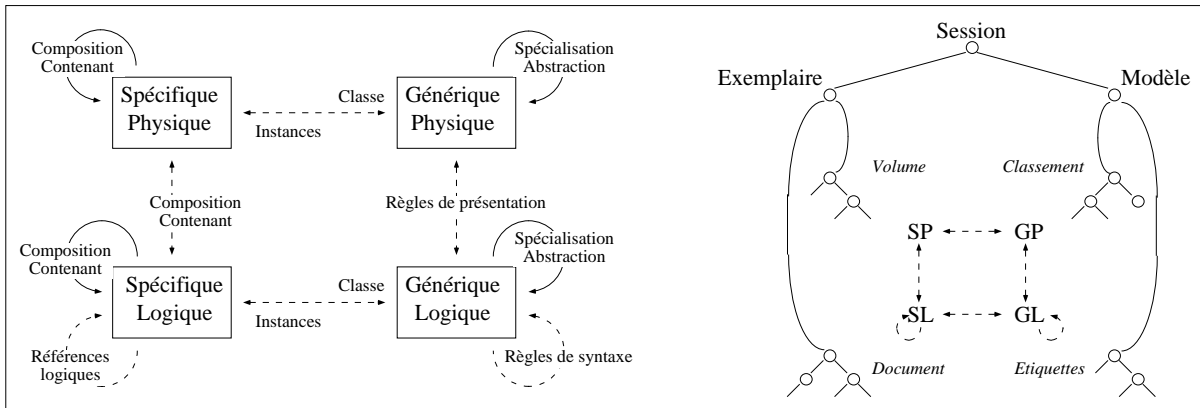


Figure 4: Relations entre entités et représentation en DAFS.

Décomposition naturelle	Image	Décomposition duale
	<p>I. Foster and S. Taylor. <i>Strand - new concepts in parallel programming</i>. Englewood cliffs, 1990.</p> <p>E. Van Herwihnen. <i>Practical SGML</i>. Kluwer Academic Publishers, 1990.</p> <p>L. Lamport. <i>Latex, a document preparation System</i>. Addison-Wesley, 1994.</p>	

Figure 5: Décompositions naturelle et duale des entités spécifiques.

- Un modèle de document définit des règles de construction qui précisent la syntaxe des éléments logiques autorisés. Ces règles syntaxiques, qu'elles prennent la forme d'une grammaire (p. ex. une DTD pour SGML) ou d'informations statistiques [6], induisent des relations de construction complexes entre entités génériques logiques. La figure 6 montre un exemple de ce type de relation, si on tente de coder chaque règle par des informations locales aux entités qui y participent.
- Un modèle de document définit également des règles de présentation. Ces directives sur le formatage s'encodent parfois via des grammaires attribuées. Dans notre modélisation des données, ces règles de présentation expriment une réécriture des entités génériques logiques en termes d'entités génériques physiques. La figure 6 montre une manière d'exprimer ces relations.

Nous avons choisi respectivement les relations compositions/contenant et les relations spéciali-

sation/abstraction pour supporter les liens hiérarchiques d'une structure DAFS. Les autres relations, en traitillé sur la figure 4, devront être simulées car elles n'appartiennent pas à l'arbre de base.

4.3 Contexte des entités

Comme nous venons de le voir, une partie du contexte des entités d'une session de reconnaissance est maintenue à travers les liens hiérarchiques de DAFS. Voici une description des autres informations à gérer localement sur les noeuds :

- Tout d'abord, il faut simuler des relations non hiérarchiques. Nous avons choisi de définir une propriété qui sert à identifier univoquement un noeud dans l'arbre. Les relations décomposition duale, classe/instances et références logiques sont directement représentées avec des propriétés qui contiennent soit un identificateur de noeud, soit une liste d'identificateurs.

Entité générique	BookTitle	gWord3
Règles de syntaxe	<i>element-of</i> : <BibRef> <i>composition</i> : Letter Letter*	–
Règles de présentation	<i>element-of</i> : <gLine2, gBlock5> <i>composition</i> : gWord3 (gWord3 gWord6)*	<i>element-of</i> : <BookTitle,...> <i>composition</i> : <Letter, ...>

Figure 6: Règles de syntaxe et de présentation, représentées localement.

Le format des règles de syntaxe et de présentation utilise aussi ces identificateurs.

- La session peut admettre des alternatives locales en remplacement des solutions partielles. DAFS nous offre deux mécanismes (noeuds OU et emprunt d’entités) pour réaliser ce concept. De plus, le choix d’une alternative peut engendrer des incohérences dans la solution globale. Nous prévoyons une propriété pour véhiculer d’éventuelles contraintes d’intégrité liées aux alternatives. Le format dépendra du moteur d’évolution de la session.
- Les entités spécifiques sont caractérisées par des attributs logiques, typographiques, ou physiques; chacun est encodé par une propriété. Les critères qui dirigent le classement hiérarchique des entités génériques physiques sont aussi gérés dans une propriété.
- Enfin, il reste à définir des propriétés pour représenter l’historique des événements subis par une entité, le taux de confiance calculé, des informations rattachées à l’interface graphique ou aux outils d’analyse.

4.4 Méthodologie d’utilisation

La définition d’une spécialisation de DAFS répond à une représentation statique d’une session de reconnaissance; nous abordons maintenant l’aspect dynamique.

L’évolution d’une session équivaut à un enrichissement incrémental de la structure DAFS. Au départ, les pages sont les seules entités spécifiques présentes; puis les différents outils d’analyse sont appelés à mesure que les informations qu’ils traitent sont prêtes. Le système procède à une unification des résultats sous DAFS, et essaie de renforcer la cohérence des zones douteuses par des révisions locales. Chaque entité physique établit un lien sur son type générique, qui en tient compte pour organiser le classement. Parallèlement à la reconstruction de la structure physique, des entités logiques sont créées

soit par étiquetage manuel, soit sur proposition de l’outil de reconnaissance; le modèle correspondant s’ajuste automatiquement pour incorporer les nouvelles instances. Ces connaissances inférées dans le modèle améliorent à leur tour les résultats de reconnaissance logique.

Ce scénario très grossier montre déjà la nécessité d’une révision des structures de contrôle dans le moteur d’une session [14] [8]. Nous tenons à exprimer isolément le comportement de chaque type d’entité, en cherchant ainsi à déléguer le contrôle à des processus concurrents; ces agents-données peuvent devenir clients d’une panoplie de spécialistes offrant des services plus ou moins raffinés sur tout le spectre des traitements automatiques (segmentation, OCR, etc.). Cette modélisation multi-agents ainsi que l’architecture logicielle à mettre en place sont exposées dans un autre article [4]. Pour valider notre approche, nous avons développé un prototype pour une application restreinte — la reconstruction complète de la structure physique. Parmi ses originalités, on peut citer le recours à la programmation multi-thread (gestion concurrente et partagée de la solution DAFS courante) et distribuée (parallélisation des traitements coûteux), ainsi qu’un couplage expressif entre moteur d’évolution et interface homme-machine.

5 Conclusion

Le projet *CIDRE* lance de nouveaux défis autour de l’architecture d’un système de reconnaissance. Dans le développement d’une plateforme logicielle pertinente, la modélisation des données joue un rôle important, plus spécialement lorsqu’on envisage de réviser le moteur d’évolution d’une session de reconnaissance en recourant à des techniques multi-agents. Nous avons proposé une manière possible de procéder à l’intégration des informations; notre solution a l’avantage de respecter les relations naturelles qu’entretiennent les différentes entités dans un contexte de reconnaissance. En pratique, nous nous appuyons sur

le format DAFS, pour lequel des outils sont distribués gratuitement et qui possède des atouts pour devenir un standard.

Les bases de notre format étant lancées, il reste à détailler la représentation exacte de certaines informations telles que les attributs typographiques. Nos outils d'analyse automatique sont maintenant presque entièrement interfacés avec ces nouvelles structures de données. Les travaux en cours portent d'une part sur le raffinement de notre prototype pour favoriser sous différentes formes le phénomène de coopération, d'autre part sur l'exploitation de nouvelles techniques en gestion de modèles de documents [6] pour faciliter à la fois l'apprentissage incrémental et la reconnaissance de structures logiques.

Références

- [1] Akindede (Oluwatoyin). – *Vers un système de construction automatique de modèles génériques de documents*. – Thèse de PhD, CRIN-Nancy, Janvier 1995.
- [2] Azokly (Antoine). – *Une approche générique pour la reconnaissance de la structure physique de documents composites*. – Thèse de PhD, IIUF-Université de Fribourg, 1995. n. 1105.
- [3] Bapst (Frédéric), Brugger (Rolf) et Ingold (Rolf). – *Towards an interactive document recognition system*. – Rapport technique n° 95-09, IIUF-Université de Fribourg, March 1995.
- [4] Bapst (Frédéric), Brugger (Rolf), Zramdini (Abdelwahab) et Ingold (Rolf). – *Integrated multi-agent architecture for assisted document recognition*. IIUF-Université de Fribourg. – 1996. soumis à DAS'96.
- [5] Bippus (Rolf), Märgner (Volker) et Schütz (Eckart). – *An interactive document segmentation and labeling system based on a universal data structure*. In : *DAS'94*. – 1994.
- [6] Brugger (Rolf), Bapst (Frédéric), Zramdini (Abdelwahab) et Ingold (Rolf). – *A heuristic approach to document structure modeling and recognition*. IIUF-Université de Fribourg. – 1996. soumis à DAS'96.
- [7] Chenevoy (Yannick). – *Reconnaissance structurelle de documents imprimés : études et réalisations*. – Thèse de PhD, CRIN-Nancy, Décembre 1993.
- [8] Cochard (Jean-Luc) et Froidevaux (Philippe). – *Environnement multi-agents de reconnaissance automatique de la parole en continu*. In : *3èmes journées francophones sur l'intelligence artificielle distribuée et les systèmes multi-agents*. – Chambéry - St-Bas-dolph, France, March 1995.
- [9] ExperVision, Inc., 3590 North First Street, San Jose, CA 95134-9815. – *TypeReader Professional*, February 1995. Release 1.0 for MacOS.
- [10] Herwijnen (Eric Van). – *Practical SGML*. – Kluwer Academic Publishers, 1990.
- [11] Hu (Tao). – *New methods for robust and efficient recognition of the logical structures in documents*. – Thèse de PhD, IIUF-Université de Fribourg, 1994. n. 1076.
- [12] Ingold (Rolf). – *Une nouvelle approche de la lecture optique intégrant la reconnaissance des structures de documents*. – Thèse de PhD, EPFL, Lausanne, 1988.
- [13] Lefèvre (Philippe) et Reynaud (François). – *Odil : an sgml description language of the layout of documents*. In : *ICDAR'95*, pp. 480–488. – 1995.
- [14] Parmentier (François) et Belaid (Abdel). – *Bibliography references validation using emergent architecture*. In : *ICDAR'95*, pp. 532–535. – 1995.
- [15] Quint (Vincent). – *Une approche de l'édition structurée des documents*. – Thèse, Université scientifique, technologique et médicale de Grenoble, 1987.
- [16] Quint (Vincent), Richy (Hélène), Roisin (Cécile) et Vatton (Irène). – *Thot - Manuel utilisateur*. – Imag - INRIA, November 1995. V0.9, successeur de Grif.
- [17] RAF Technology, Inc. – *DAFS library - Programmer's guide and reference*, August 1995. <http://documents.cfar.umd.edu/>.
- [18] Zramdini (Abdelwahab). – *Study of optical font recognition based on global typographical features*. – Thèse de PhD, IIUF-Université de Fribourg, 1995. n. 1106.