



# Reconnaissance à la volée de documents structurés manuscrits en-ligne

Sébastien Macé, Eric Anquetil

► **To cite this version:**

Sébastien Macé, Eric Anquetil. Reconnaissance à la volée de documents structurés manuscrits en-ligne. Actes du 9ème Colloque International Francophone sur l'Écrit et le Document Colloque International Francophone sur l'Écrit et le Document, Sep 2006, France. pp.175-180. hal-00113589

**HAL Id: hal-00113589**

**<https://hal.archives-ouvertes.fr/hal-00113589>**

Submitted on 13 Nov 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Reconnaissance à la volée de documents structurés manuscrits en-ligne

Sébastien Macé – Éric Anquetil

IRISA - INSA  
Campus Universitaire de Beaulieu  
35042, Rennes Cedex, France

{Sebastien.Mace, Eric.Anquetil}@irisa.fr

**Résumé** : Dans ce papier, une nouvelle approche pour l'interprétation de documents structurés manuscrits en-ligne est présentée. Elle est basée sur un formalisme flexible et générique permettant la reconnaissance à la volée des éléments d'un document structuré. L'originalité du formalisme est la modélisation du couplage d'une vision globale du document analysé avec une vision locale de l'élément à reconnaître. L'analyseur pilote alors des reconnaissseurs de formes dédiés en fonction du contexte structurel de l'élément analysé. Nous détaillons plus particulièrement le processus de prise de décision en cas d'ambiguïté entre plusieurs interprétations possibles. Nous exploitons la théorie des sous-ensembles flous afin de prendre en compte la nature imprécise du tracé manuscrit et des contextes structurels modélisés. Cette approche a été validée avec le développement de trois systèmes orientés stylo : pour l'édition de partitions musicales, de graphes et de diagrammes de classes UML.

**Mots-clés** : Reconnaissance en-ligne, interprétation de documents structurés, interprétation en contexte, théorie des sous-ensembles flous.

## 1 Introduction

Alors que l'interaction homme-machine orientée stylo est en pleine expansion, les systèmes qui en tirent pleinement parti afin de permettre la réalisation en-ligne de documents structurés à main levée, directement saisis sur un écran tactile, restent limités. Nous définissons comme *documents structurés* des documents dont les éléments obéissent à des règles de structuration, comme par exemple des schémas électriques, des partitions musicales, des diagrammes, des plans, etc. L'interprétation de tracés manuscrits, appelée *interprétation en-ligne*, est un problème encore plus complexe dans le cadre de l'analyse de tels documents car ils sont constitués d'éléments de natures différentes qui ne peuvent pas tous être interprétés à l'aide d'un seul reconnaissseur de formes manuscrites. De plus, l'interprétation d'un tracé peut dépendre du contexte dans lequel il a été réalisé ; ce contexte doit donc être pris en compte.

L'interprétation *a posteriori*, qui consiste à analyser un document structuré en-ligne une fois sa composition terminée [NAK 93], est une approche prometteuse pour offrir une interprétation non-contrainte des tracés manuscrits. Mais les difficultés pour réaliser une analyse *a posteriori* automatique

font qu'il s'agit toujours aujourd'hui d'un problème ouvert. L'interprétation à la volée, qui consiste à analyser les tracés au fur et à mesure de leur composition [BLO 02], est une autre manière de considérer l'interprétation de documents structurés en-ligne. Le système doit alors analyser les tracés manuscrits au fur et à mesure de leur composition et prendre une décision aussi rapidement que possible afin de ne pas faire attendre l'utilisateur. Le but de ce papier est de proposer une méthode générique pour reconnaître à la volée des documents structurés manuscrits en-ligne.

La plupart des approches développées pour l'analyse à la volée de documents structurés manuscrits sont principalement *ad hoc* et dédiées à un type particulier de documents. Quelques auteurs ont proposé des méthodes génériques, principalement basées sur la description des éléments du document [ALV 05]. Cependant, ces descriptions sont locales à l'élément analysé et ne tiennent pas compte de la structure globale du document. Enfin, ils ne permettent pas l'interprétation d'éléments trop complexes, comme certains symboles ou du texte, qui peuvent être reconnus par des reconnaissseurs de formes puissants basés par exemple sur des réseaux de neurones ou sur des chaînes Markov cachées.

Notre approche pour reconnaître des formes en contexte est basée sur un formalisme générique qui modélise comment interpréter les différents éléments d'un document structuré. Son originalité est le couplage de l'exploitation de reconnaissseurs de formes classiques avec une description du *contexte structurel* que doit vérifier l'élément analysé, *i.e.* comment il doit être positionné relativement aux autres éléments du document. Ce formalisme ayant déjà été présenté dans [MAC 06a, MAC 06b], nous nous focalisons ici plus en détail sur l'évaluation de l'appartenance à des contextes structurels. Celle-ci est basée sur la théorie des sous-ensembles flous, de manière à prendre en compte la nature imprécise du tracé manuscrit et de l'emplacement de ces contextes. Cette évaluation permet de prendre une décision en cas d'ambiguïté, *i.e.* lorsque plusieurs interprétations différentes sont possibles. L'approche proposée a été en particulier appliquée pour l'interprétation à la volée de partitions musicales, de graphes et de diagrammes de classes UML.

Dans la section suivante, nous introduisons le formalisme permettant de modéliser l'interprétation à la volée de documents structurés manuscrits. Nous nous focalisons ensuite

dans la section 3 sur la notion d'appartenance à des contextes structurels, qui est basée sur la théorie des sous-ensembles flous, et expliquons comment elle est exploitée dans le processus d'interprétation et dans la gestion des ambiguïtés. Dans la section 4 nous présentons quelques résultats et enfin nous concluons et donnons quelques pistes de travaux futurs.

## 2 Un formalisme pour l'interprétation de documents structurés

Dans [MAC 05, MAC 06b], nous avons défini les besoins fondamentaux que nous estimons associés à la modélisation de l'interprétation de documents structurés manuscrits en-ligne : *la gestion des informations chronologiques, la représentation de la structure physique du document, le pilotage du processus de reconnaissance par une analyse du contexte structurel du document et l'interaction homme-machine orientée stylo*. L'approche proposée est basée sur un formalisme qui prend en compte chacun de ces concepts pour modéliser comment interpréter à la volée les éléments de documents structurés manuscrits. Il permet l'écriture d'un ensemble de règles d'interprétation, chacune définissant comment l'élément dont elle porte le nom peut être interprété et généré. Nous utilisons le terme *élément* pour tout ce que peut contenir un document, c'est-à-dire aussi bien pour un tracé manuscrit que pour un élément du document qui a déjà été interprété auparavant. Etant donné que plusieurs règles peuvent porter le même nom, il est possible de modéliser plusieurs manières d'interpréter un même élément, et donc plusieurs manières de le composer.

Une règle prend un ensemble d'éléments en paramètre, et crée un nouvel élément qui peut remplacer ces paramètres dans le document en cours de composition ; les paramètres sont donc les constituants du nouvel élément, ce qui permet de décrire de manière hiérarchique des éléments composés.

Une règle est constituée de quatre blocs différents : le bloc de *vérification de contextes de document (VCD)*, le bloc de *vérification de contextes de forme (VCF)*, le bloc de *reconnaissance de forme (RF)* et le bloc de *création de contextes de document (CCD)*. Elle est de la forme :

**NomDuSymbole** (*Paramètre 1*, ..., *Paramètre N*)  
 bloc de *vérification de contextes de document (VCD)*.  
 bloc de *vérification de contextes de forme (VCF)*.  
 bloc de *reconnaissance de forme (RF)*.  
 bloc de *création de contextes de document (CCD)*.

D'une part, les blocs *VCD* et *CCD* permettent d'avoir une *vision globale* du document en cours de composition afin de modéliser dans quels contextes structurels du document un élément peut ou bien doit exister. D'autre part, les blocs *VCF* et *RF* permettent d'avoir une *vision locale* de l'élément à reconnaître, en fonction de ces contextes structurels. Avant de présenter formellement chacun des quatre blocs, la prochaine sous-section introduit le processus d'analyse exploitant les connaissances modélisées à l'aide du formalisme.

### 2.1 Présentation du processus d'analyse

Chaque fois que l'utilisateur dessine un tracé, le processus d'analyse est exploité. Comme présenté figure 1, le but est de déterminer quelle suite de règles appliquer. Cette recherche est guidée par les contextes structurels que contient

le document. Lorsqu'un nouvel élément est créé, il peut permettre l'existence d'autres éléments ; il définit alors de nouveaux contextes dans le document, spécifiant quels éléments peuvent maintenant exister à une position donnée, *i.e.* quelles règles doivent être essayées sur un tracé dessiné ici (bloc *CCD*). Ainsi, la première étape de l'analyse d'un tracé consiste à déterminer dans lesquels de ces contextes il se situe et à activer les règles associées. Ainsi, seules les règles pertinentes sont appelées : nous n'essayons pas de reconnaître un élément qui ne peut pas exister dans le contexte dans lequel il a été dessiné. De plus, pour qu'une règle soit activée, tous ses paramètres doivent déjà exister.

Chaque règle spécifie les contextes structurels dans lesquels le nouvel élément doit exister (bloc *VCD*) ainsi que des contraintes locales que ses paramètres doivent satisfaire (bloc *VCF*) ; si toutes sont vérifiées, un reconnaiseur de formes dédié peut être appelé sur ces paramètres (bloc *RF*). En cas de succès, la règle est applicable. Nous ne détaillons pas ici le processus de choix de la règle appliquée lorsque plusieurs sont applicables, puisqu'il fait l'objet de la section 3. En conséquence de l'application d'une règle, le nouvel élément remplace ses paramètres dans le document et de nouveaux contextes structurels peuvent être créés (bloc *CCD*).

Si aucune règle n'est applicable sur un tracé manuscrit, il est rejeté et disparaît du document. Si une règle est appliquée, le nouvel élément correspondant est créé, ce qui termine l'itération courante du processus d'interprétation ; une nouvelle itération commence alors sur le même schéma, cherchant cette fois à appliquer des règles sur ce nouvel élément. Ceci permet de déterminer si le nouvel élément interagit avec d'autres éléments du document afin de former un élément plus complexe.

Pour finir, nous exploitons l'interaction homme-machine existant dans le cadre d'une analyse à la volée et intégrons l'utilisateur dans le processus d'interprétation : il peut implicitement valider une réponse du système en continuant la composition du document après que celle-ci ait été affichée, ou bien explicitement la rejeter en dessinant un geste le supprimant. Ainsi, l'analyseur n'a pas besoin de remettre en cause une décision prise auparavant, car elle a été validée par l'utilisateur. Le processus d'analyse est donc plus robuste.

La figure 1 récapitule le mécanisme global d'analyse des éléments du document.

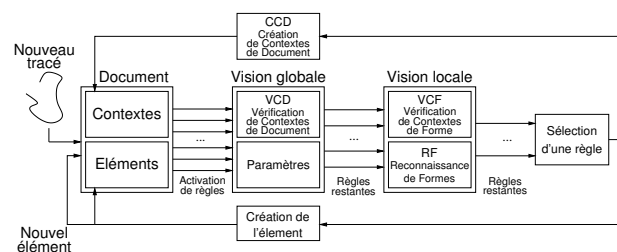


FIG. 1 – Illustration du mécanisme d'analyse à la volée des éléments d'un document structuré manuscrit.

Les prochaines sous-sections présentent la modélisation de la connaissance exploitée par l'analyseur. Nous illustrons les notations sur des exemples de symboles constituant une application simple de composition de graphes (*c.f.* figure 6).

## 2.2 Les blocs de vérification et de création de contextes de document

Le but du couplage des blocs *VCD* et *CCD* est la modélisation de la structure physique d'un document, représentée par un ensemble de contextes structurels, permettant d'en avoir une vision globale. Un contexte structurel définit un emplacement particulier du document.

- *Le bloc VCD* est constitué d'une liste de contextes structurels qui doivent être vérifiés par des éléments du document pour qu'une règle soit applicable ; il modélise donc les contextes structurels dans lesquels l'élément peut être créé par la règle. Ce bloc fait référence à des contextes générés par un bloc *CCD*. Sa syntaxe est la suivante :

$$VCD : R_1[position_1, partie_1]A_1 , \\ \dots , \\ R_N[position_N, partie_N]A_N .$$

Cette expression signifie qu'à la position relative *position<sub>i</sub>* (par exemple à *gauche*, *au dessus*, *etc.*) d'un élément *R<sub>i</sub>*, la partie *partie<sub>i</sub>* (par exemple *tous les points*, *un*, *le premier*, *le plus à gauche*, *etc.*) d'un élément *A<sub>i</sub>* doit exister. *R<sub>i</sub>* et *A<sub>i</sub>* ne sont pas obligatoirement des paramètres de la règle. Par exemple, dans un graphe, une connexion doit être dessinée de façon à ce que son premier et son dernier points soient dans un sommet ; un bloc *VCD* de règle prenant un tracé *t* en paramètre et modélisant ceci peut être :

$$VCD : Sommet[dans, premier]t , \\ Sommet[dans, dernier]t .$$

- *Le bloc CCD* spécifie les contextes structurels de document qui sont générés en conséquence de la reconnaissance d'un élément et qui permettront d'activer les règles correspondantes. Sa syntaxe est la suivante :

$$CCD : R_1[position_1, partie_1]A_1^{n_1} , \\ \dots , \\ R_N[position_N, partie_N]A_N^{n_N} .$$

Ceci signifie qu'à la position relative *position<sub>i</sub>* d'un élément *R<sub>i</sub>*, la partie *partie<sub>i</sub>* d'un élément *A<sub>i</sub>* peut exister. L'élément en cours de création peut être référencé par *this*. Le nombre *n<sub>i</sub>* indique la quantité d'éléments *A<sub>i</sub>* qui peuvent exister à cet emplacement et vaut \* si il n'y a pas de limite. Par exemple, la création d'un sommet rend possible le dessin d'un numéro dedans, ainsi que celui de connexions qui en partent ou y arrivent ; un bloc *CCD* modélisant ceci peut être :

$$CCD : this[dans, tous]Numéro^1 , \\ this[dans, premier]Connexion^* , \\ this[dans, dernier]Connexion^* .$$

## 2.3 Les blocs de vérification de contextes et de reconnaissance de formes

Le but du couplage des blocs *VCF* et *RF* est une vision locale d'un élément à reconnaître ; il distribue le processus de reconnaissance sur des contraintes locales (bloc *VCF*) et des reconnaisseurs de formes (bloc *RF*).

- *Le bloc VCF* est une liste de contraintes locales qui doivent être vérifiées par les paramètres de la règle afin de reconnaître un élément constitué de plusieurs sous-éléments : il permet d'identifier quel constituant du nouvel élément correspond à chacun des paramètres. La syntaxe est la même que celle du bloc *VCD*, mais cette fois les éléments mis en jeu sont des paramètres de la règle. Il n'y a par défaut aucune contrainte d'ordre de composition de ces sous-éléments ; si c'est nécessaire, un opérateur chronologique peut être utilisé. Par

exemple, un rectangle, qui est un sommet dans notre cadre de composition de graphes, est constitué de deux segments horizontaux, notés *sh1* et *sh2*, et de deux segments verticaux, notés *sv1* and *sv2* ; il ne semble pas pertinent d'imposer un ordre dans lequel les réaliser. Ainsi, un bloc *VCF* peut être :

$$VCF : sh1[droite, haut]sv1 , \\ sv1[bas, droite]sh2 , \\ sh2[gauche, bas]sv2 , \\ sv2[haut, gauche]sh1 .$$

Ces contraintes locales ne sont pas nécessairement suffisantes pour assurer que les segments forment un rectangle ; nous pouvons alors juger utile d'exploiter, en complément de cette description, un système de reconnaissance de formes classique. Ceci peut être réalisé à l'aide du bloc *RF*.

- *Le bloc RF* correspond à un appel à un reconnaisseur de formes et permet, en fonction du contexte d'un élément, de n'appeler que les reconnaisseurs pertinents ; ce mécanisme est essentiel afin d'augmenter la fiabilité du processus d'interprétation, car moins un reconnaisseur a de classes de formes à discriminer, et plus il est efficace et fiable. L'ordre dans lequel les éléments sont envoyés au reconnaisseur correspond à l'ordre dans lequel sont énoncés les paramètres ; ainsi, il est toujours le même, ce qui soulage également le travail du reconnaisseur. Notons qu'afin de limiter les ambiguïtés, nous exploitons des reconnaisseurs fonctionnant avec rejet [MOU 06] : ils ne donnent une solution que si leur confiance en celle-ci est suffisante. La syntaxe du bloc *RF* est la suivante :

$$RF : FamilleDeFormes , \{RéponsesAcceptées\} .$$

Ceci signifie que le reconnaisseur dédié à la famille de formes *FamilleDeFormes* est appelé sur les paramètres de la règle ; si la réponse est incluse dans *RéponsesAcceptées*, alors le processus de reconnaissance réussit. Par exemple, un bloc *RF* pour la règle **Rectangle** peut être :

$$RF : FormesGéométriques , \{Rectangle\} .$$

Pour finir, voici deux règles d'interprétation complémentaires permettant de générer des éléments de type **Rectangle**, l'une pour un rectangle dessiné avec un seul tracé (règle du haut), l'autre pour un rectangle dessiné avec quatre tracés (règle du bas). Notons que dans la première, tout le processus de reconnaissance de la forme est réalisé par un reconnaisseur de formes, alors que dans la seconde, il est réalisé par les contraintes locales.

**Rectangle** (*Trace t*)

$$VCD : aucun .$$

$$VCF : aucun .$$

$$RF : FormesGéométriques , \{Rectangle\} .$$

$$CCD : this[dans, tous]Poids^1 , \\ this[dans, premier]Connexion^* , \\ this[dans, dernier]Connexion^* .$$

**Rectangle** (*SegH sh1 , SegV sv1 , SegH sh2 , SegV sv2*)

$$VCD : aucun .$$

$$VCF : sh1[droite, haut]sv1 , sv1[bas, droite]sh2 , \\ sh2[gauche, bas]sv2 , sv2[haut, gauche]sh1 .$$

$$RF : aucun .$$

$$CCD : this[dans, tous]Poids^1 , \\ this[dans, premier]Connexion^* , \\ this[dans, dernier]Connexion^* .$$

## 3 Evaluation floue des contextes structurels

La nature imprécise du tracé manuscrit ne se manifeste pas uniquement dans sa forme, mais également dans l'empla-

cement auquel il est réalisé. Considérons les éléments présentés sur la figure 2, dans le cadre de l'interprétation de rectangles dessinés à l'aide de quatre segments. Dans chacun des cas, l'utilisateur dessine un tracé dans un document contenant déjà six autres éléments qui sont des segments. Ce tracé est interprété comme un segment lors de la première itération du processus d'interprétation ; ce nouveau segment devient l'élément analysé dans une seconde itération. Il s'agit alors de déterminer si la règle **Rectangle** est applicable, *i.e.* si ce nouveau segment interagit avec d'autres éléments du document pour former un rectangle. Il semble naturel de ne pas faire les mêmes choix dans les trois cas présentés. En (1), le tracé devrait s'associer avec les trois segments de gauche pour former un rectangle ; en (2), il semble aussi naturel de l'associer à ceux de gauche qu'à ceux de droite, ce qui rend toute décision impossible et autorise un rejet du tracé sans même l'interpréter comme un segment (rejetant donc également le résultat de la première itération) ; pour finir, en (3), le nouvel élément paraît trop éloigné des segments pour s'associer à eux afin de former un rectangle : la seconde itération échoue et le tracé devient un segment, qui pourra par exemple être le début d'un rectangle.

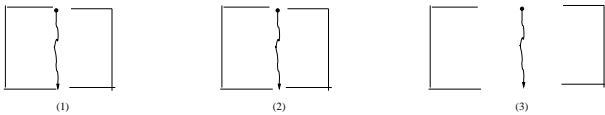


FIG. 2 – Trois tracés manuscrits qui devraient produire des interprétations différents.

Cet exemple met également en valeur le fait qu'il n'est pas possible de prendre une décision de type "tout ou rien" en ce qui concerne la vérification d'un contexte structurel ; la théorie des sous-ensembles flous est alors une approche intéressante permettant de prendre en compte l'imprécision de telles relations spatiales. L'approche que nous proposons s'inspire en partie de celle de Bloch *et al.* [BLO 99]. Pour chaque contexte, nous définissons un "paysage flou" modélisant pour tout point son degré d'appartenance à ce contexte ; ensuite, nous évaluons avec quel degré un objet analysé composé de plusieurs points appartient à ce paysage flou. Cette valeur correspond au degré de vérification par cet objet de la relation en cours d'analyse.

Dans la prochaine sous-section, nous présentons comment définir un paysage flou, puis nous montrons comment évaluer l'appartenance d'un élément à un tel sous-ensemble flou.

### 3.1 Définition d'un paysage flou $\mu_C$

Les contextes structurels énoncés dans les règles d'interprétation sont définis *a priori*, à l'aide d'experts du domaine correspondant à l'application. Ces emplacements sont délimités de façon stricte, et nous les exploitons afin de définir le paysage flou  $\mu_C$  de chaque contexte structurel  $C$  : ils correspondent au noyau du sous-ensemble flou (points pour lesquels le degré d'appartenance vaut 1) autour duquel la fonction d'appartenance diminue linéairement avec l'augmentation de la distance à ce noyau, jusqu'aux limites du support (points pour lesquels le degré d'appartenance n'est

pas nul). La figure 3 présente à gauche une projection sur une dimension de la fonction d'appartenance d'un point au paysage flou, produisant une courbe trapézoïdale, et à droite deux exemples de fonctions d'appartenance en deux dimensions qui ont des noyaux de formes différentes et qui donc produisent des paysages flous différents.

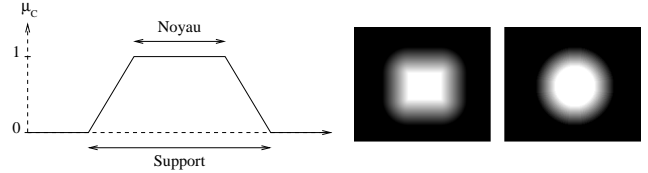


FIG. 3 – Exemple de fonction d'appartenance d'un point au paysage flou en une dimension (gauche), et deux exemples en deux dimensions pour deux noyaux de formes différentes (droite, les hautes valeurs de gris correspondent aux degrés d'appartenance élevés).

### 3.2 Définition du degré d'appartenance d'un élément à un paysage flou

Rappelons que nous notons un contexte structurel  $C$  par  $R[\text{position}, \text{partie}]$ . Notons  $A$  l'objet attendu dans ce contexte structurel. Le but est de déterminer avec quel degré un objet analysé  $A$  satisfait un contexte structurel  $C$  défini par un paysage flou et donc par une fonction d'appartenance  $\mu_C$ .  $A$  est ici un élément du document, par exemple un tracé. Comme ces éléments sont constitués de points, il est naturel d'attribuer un degré d'appartenance à l'objet analysé de 1 pour chacun de ces points, et de 0 pour les autres.

Lorsque le contexte structurel ne met en jeu qu'un point particulier de l'élément analysé (*i.e.* si *partie* vaut *le premier point*, *le point le plus à gauche*, *etc.*), le degré d'appartenance de l'élément au contexte structurel correspond à celui du point considéré.

Si au moins un point quelconque de l'élément doit vérifier le contexte (*i.e.* si *partie* vaut *un*), le degré d'appartenance de l'élément au contexte structurel correspond à celui du point qui le vérifie le mieux :

$$\mu_C(A) = \max_{x \in A} \mu_C(x) \quad (1)$$

Si tous les points doivent être pris en compte (*i.e.* si *partie* vaut *tous*), nous calculons une valeur moyenne qui fournit une évaluation globale de l'appartenance de  $A$  au sous-ensemble flou  $\mu_C$  [BLO 99]. Cette valeur moyenne peut être définie par l'équation suivante ( $|A|$  désigne le nombre de points de l'élément analysé) :

$$\mu_C(A) = M_C(A) = \frac{1}{|A|} \sum_{x \in A} \mu_C(x) \quad (2)$$

### 3.3 Fusion de fonctions d'appartenance à des contextes structurels

Nous présentons dans cette sous-section comment fusionner les degrés d'appartenance de tous les contextes structurels d'une règle afin de donner un degré de cohérence "globale" de ses contextes.

Les contextes structurels d'une règle  $R$  sont dans le bloc  $VCD$  (contextes notés  $C_{VCD}^1, \dots, C_{VCD}^N$ ) et dans le bloc  $VCf$  (contextes notés  $C_{VCf}^1, \dots, C_{VCf}^{N'}$ ). Nous jugeons que tous ces contextes jouent un rôle équivalent dans la vérification de la règle. Nous fusionnons leur degré d'appartenance à l'aide d'un opérateur de t-norme (conjonction floue) afin de définir le degré  $\mu_R$  de cohérence "globale" des contextes structurels d'une règle de la façon suivante (les  $A_X^Y$  désignent les objets analysés des contextes étudiés) :

$$\mu_R = \text{t-norme}(\mu_{C_{VCD}^1}(A_{VCD}^1), \dots, \mu_{C_{VCD}^N}(A_{VCD}^N), \mu_{C_{VCf}^1}(A_{VCf}^1), \dots, \mu_{C_{VCf}^{N'}}(A_{VCf}^{N'})). \quad (3)$$

En pratique, notre opérateur de t-norme est la multiplication. Il est nécessaire de normaliser le résultat obtenu afin de pouvoir ensuite comparer le degré de vérification de plusieurs règles, et en particulier afin de ne pas favoriser des règles qui ont moins de contextes à vérifier. Nous normalisons donc par le nombre de contextes de la règle et déduisons le degré de cohérence "globale" des contextes d'une règle :

$$\mu_R^{norm} = \left( \prod_{0 \leq d \leq N} \mu_{C_{VCD}^d}(A_{VCD}^d) \cdot \prod_{0 \leq f \leq N'} \mu_{C_{VCf}^f}(A_{VCf}^f) \right)^{\frac{1}{N+N'}}. \quad (4)$$

La figure 4 présente l'impact du calcul du degré de cohérence "globale" que nous proposons : si un contexte est peu vérifié, il diminue le degré de vérification maximum possible de la règle. Cette figure présente l'impact de la localisation des premier et dernier points d'un segment pour reconnaître un rectangle en fonction des éléments déjà existants. Sur la gauche de la figure, les segments se touchent presque, ce qui facilite la reconnaissance d'un rectangle dans le cas de l'ajout d'un segment vertical ; au contraire, à droite les segments sont plus éloignés, ce qui diminue par avance le degré maximum de vérification des contextes qui peut être obtenu suite à la réalisation d'un quatrième tracé.

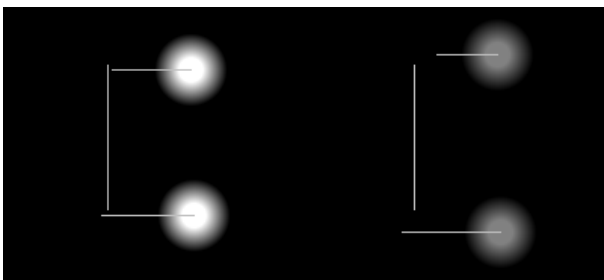


FIG. 4 – Impact du contexte global de la forme sur le calcul de la vérification des contextes structurels.

### 3.4 Mécanisme de prise de décision

Lorsque plusieurs règles d'interprétation sont applicables, nous prenons en compte le degré de vérification de leurs contextes structurels : la règle qui a le degré le plus élevé est la règle choisie.

Cependant, nous avons vu qu'il existe des situations d'ambiguïté dans lesquelles plusieurs décisions sont pos-

sibles mais la différence entre ces deux solutions est tellement faible qu'il ne semble pas pertinent d'en choisir une en particulier. Le bon choix consiste alors à rejeter toute interprétation. Pour cela, nous exploitons les définitions concernant le *rejet de confusion* utilisé dans le cadre de la reconnaissance de formes : il permet de rejeter une forme si le reconnaisseur hésite entre deux classes, *i.e.* si la différence entre les deux meilleures classes n'est pas assez grande [MOU 06]. Nous transposons ces définitions pour la recherche du contexte structurel le plus approprié, et comparons donc les scores des deux meilleures règles, notés  $R1$  et  $R2$ , tels que  $\mu_{R1}^{norm} \geq \mu_{R2}^{norm}$ . Nous définissons une fonction de confiance de la manière suivante :

$$\psi_{R1,R2} = \frac{\mu_{R1}^{norm} - \mu_{R2}^{norm}}{\mu_{R1}^{norm}} \quad (5)$$

La politique choisie consiste à fixer un seuil limite pour cette fonction, au dessous duquel il n'est pas cohérent de prendre une décision. Nous avons expérimentalement fixé ce seuil à 0.05.

Le tableau 1 regroupe les calculs des degrés de cohérence "globale" des règles d'interprétation **Rectangle** sur le quatre segments de gauche (notée  $\mu_{Rectangle1}^{norm}$ ) et sur les quatre de droite (notée  $\mu_{Rectangle2}^{norm}$ ) de la figure 2, la valeur de la fonction de confiance correspondante, ainsi que la décision prise en conséquence de ces calculs. Nous pouvons voir que les résultats correspondent à ceux que nous attendons.

	Fig. 2.(1)	Fig. 2.(2)	Fig. 2.(3)
$\mu_{Rectangle1}^{norm}$ (gauche)	0.95	0.83	0.00
$\mu_{Rectangle2}^{norm}$ (droite)	0.65	0.81	0.00
$\psi_{Rectangle1,Rectangle2}$	0.32	0.02	-
Décision prise	Rectangle de gauche	Rejet du tracé	Segment (pas de rectangle)

TAB. 1 – Décisions prises pour chacun des trois exemples donnés figure 2.

La figure 5 permet de visualiser la prise de décision entre deux règles en concurrence : il s'agit de déterminer auquel des deux segments déjà existant dans le document un nouveau tracé serait connecté. Nous pouvons voir d'une part que plus on s'éloigne d'une extrémité de segment et plus le degré de vérification de la relation diminue, et d'autre part qu'une zone de rejet apparaît au niveau de l'intersection des deux contextes.

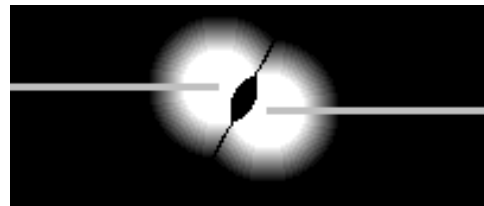


FIG. 5 – Paysage flou permettant la visualisation de la prise de décision et de la zone de rejet.

## 4 Mise en oeuvre du système proposé

Afin de mettre en valeur l'aspect générique de notre approche, nous présentons trois exemples de systèmes qui ont été développés sur tablet PC à l'aide de la méthode présentée. Le développement d'un système basé sur une interaction stylo avec notre approche ne nécessite que l'écriture des règles d'interprétations, le développement des reconnaissseurs correspondants basés ici sur des réseaux de neurones à fonctions à base radiale (RBFN) et la spécification de l'aspect graphique de chacun des éléments du document.

La figure 6 présente en haut deux captures d'écran du prototype d'éditeur de partitions musicales. Il permet le dessin des symboles musicaux majeurs (clés, notes, altérations, hampes, ligatures, barres de mesure, etc.). La figure présente au milieu deux captures d'écran de l'éditeur de graphes et en bas de l'éditeur de diagrammes de classes UML. Ce dernier permet le dessin de classes (à l'aide de rectangles), de leur nom, attributs et fonctions, et de relations telles que l'héritage, l'agrégation, etc. Dans chaque cas, nous présentons à gauche un document interprété et à droite les tracés manuscrits correspondants. Bien que cela n'apparaisse pas sur cette figure, le processus d'analyse est à la volée : les tracés de l'utilisateur sont remplacés par leur symbole correspondant au fur et à mesure de leur réalisation. Notons que nous n'avons pas encore associé l'éditeur de diagrammes UML avec un reconnaissseur d'écriture manuscrite : les tracés correspondant à du texte sont identifiés comme tels mais pas encore interprétés. Le lecteur intéressé pourra trouver des informations plus complètes concernant ces systèmes ainsi que leurs possibilités d'édition (déplacement, suppression d'éléments, etc.) basées sur l'interaction stylo dans [MAC 06a].

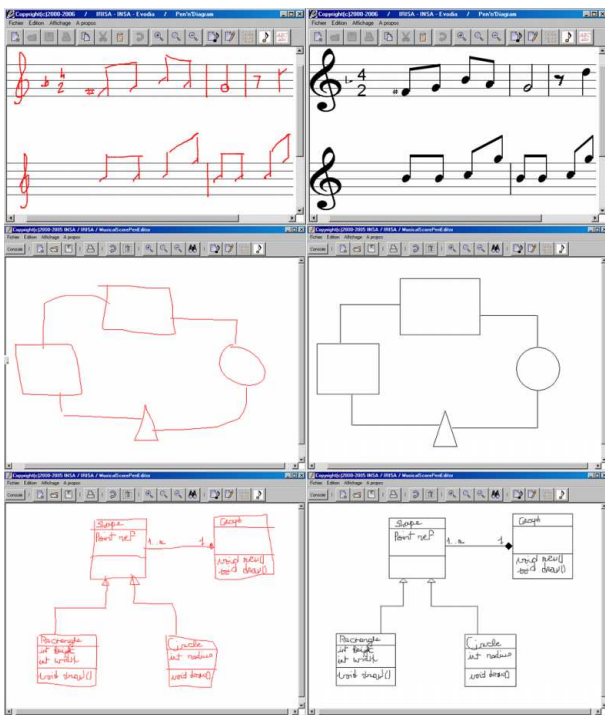


FIG. 6 – Captures d'écran des prototypes développés à l'aide de la méthode proposée.

## 5 Conclusion et travaux futurs

Dans ce papier, nous avons présenté une nouvelle approche permettant l'interprétation à la volée de documents structurés manuscrits en-ligne, basée sur le pilotage du processus d'interprétation par une analyse structurale. Nous avons mis en valeur la généricité et la flexibilité du formalisme que nous avons introduit afin de modéliser cette interprétation. Nous exploitons la théorie des sous-ensembles flous afin d'évaluer l'appartenance aux contextes structurels dans lesquels se situe un élément et de choisir l'interprétation la plus adaptée. Le système est opérationnel et a déjà été utilisé pour développer trois prototypes de systèmes orientés stylo : pour l'édition de partitions musicales, de graphes et de diagrammes de classes UML. Nos prochains travaux vont avoir pour but de combiner les résultats de l'analyse structurale du document et des reconnaissseurs de formes, de manière à proposer une approche encore plus robuste.

## 6 Remerciements

Les auteurs souhaitent remercier Guy Lorette, Professeur à l'Université de Rennes 1, pour ses précieux conseils. Ce projet bénéficie du soutien financier de la Région Bretagne.

## Références

- [ALV 05] ALVARADO C., DAVIS R., Dynamically Constructed Bayes Nets for Multi-Domain Sketch Understanding, *Proc. of IJCAI 2005*, 2005, pp. 1407-1412.
- [BLO 99] BLOCH I., Fuzzy Relative Position between Objects in Image Processing : a Morphological Approach, *IEEE Transactions on PAMI*, vol. 21, n° 7, 1999, pp. 657-664.
- [BLO 02] BLOSTEIN D., LANK A., ROSE A., ZANIBBI R., *Graphics Recognition : Algorithms and Applications*, vol. 2390, Chapitre User Interfaces for On-Line Diagram Recognition, pp. 92-103, LNCS, 2002.
- [MAC 05] MACÉ S., ANQUETIL E., COÛASNON B., A generic method to design pen-based systems for structured document composition : Development of a musical score editor, *Proc. of the 1st Workshop on Improving and Assessing Pen-Based Input Techniques*, 2005, pp. 15-22.
- [MAC 06a] MACÉ S., ANQUETIL E., *Computer-Aided Design of User Interfaces V, Proc. of CADUI'06*, Chapitre A Generic Approach for Pen-based User Interface Development, pp. 57-70, Springer-Verlag, 2006.
- [MAC 06b] MACÉ S., ANQUETIL E., A Generic Approach for Eager Interpretation of On-line Handwritten Structured Documents, *Proc. of ICPR'06 (accepted)*, 2006.
- [MOU 06] MOUCHERE H., ANQUETIL E., RAGOT N., Étude et gestion des types de rejet pour l'optimisation de classifieurs, *Actes de RFIA 2006*, 2006.
- [NAK 93] NAKAGAWA M., MACHII K., KATO N., SOUYA T., Lazy recognition as a principle of pen interfaces, *Proc. of INTERACT'93 and CHI'93*, 1993, pp. 89-90.