ANR-PASSAGE

Livrable : L1

Titre du document : Définition du formalisme d'annotation

Numéro de référence du projet	ANR-06-MDCA-013-02
Acronyme du projet	PASSAGE
Titre du projet en entier	PASSAGE: Produire des annotations syntaxiques à grande échelle pour aller de l'avant
Client	Patrick Paroubek CNRS-LIMSI Campus universitaire BAT 508 BP 133 91404 ORSAY cedex
Sous-contractant	Gil Francopoulo TAGMATICA 126 rue de Picpus 75012 PARIS
Type de document	Rapport
Date contractuelle de livraison	31 décembre 2007

Evolution du document

version	date	version	date
1.0	31 octobre 2007	1.8	29 avril 2008
1.1	16 janvier 2008	1.9	20 mai 2008
1.2	30 janvier 2008	1.10	11 juin 2008
1.3	31 janvier 2008	1.11	23 octobre 2008
1.4	22 février 2008	1.12	20 janvier 2009
1.5	25 mars 2008	1.13	6 mars 2009
1.6	31 mars 2008	1.14	6 mai 2009
1.7	10 avril 2008		

1 Introduction

Il s'agit de définir le format d'annotation Passage.

2 Architecture

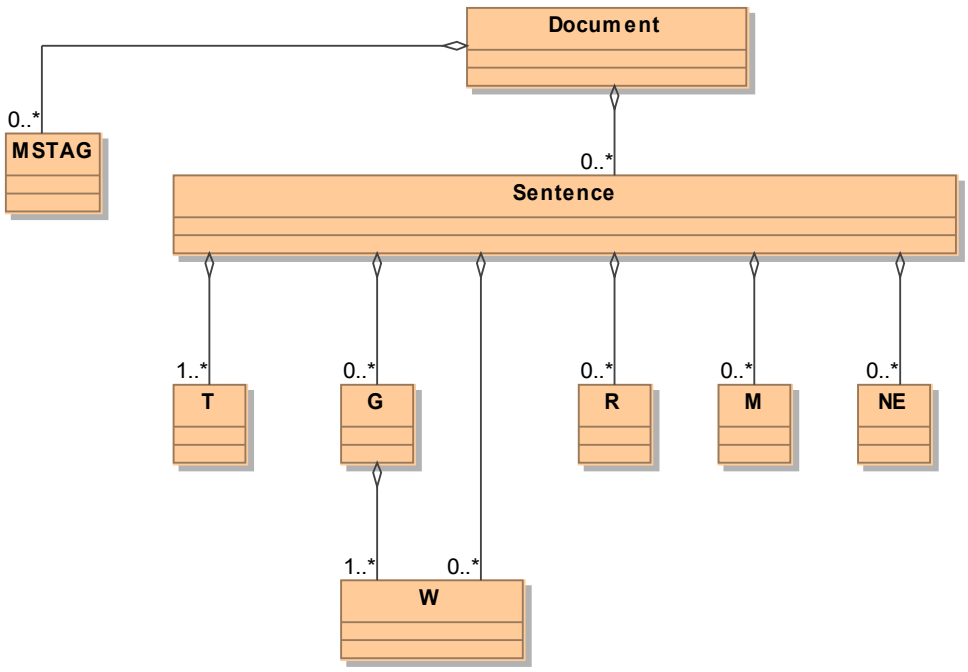
Le format est multi-niveaux de manière hiérarchique, à savoir :

- Niveau Mot Forme
- Niveau Groupe, au sens de groupe syntaxique
- Niveau Sentence
- Niveau Document

De manière non hiérarchique, nous avons :

- le token
- la relation
- la marque
- l'entité nommée
- la combinaison de traits morpho-syntaxiques

La structure est synthétisée par le diagramme de classe UML suivant [1]:



Pour simplifier la présentation, les enchâssements des relations ne sont pas dessinés, voir pour cela la DTD en annexe.

Selon les niveaux, un mécanisme de notation enchâssée (i.e. 'embedded' notation ou 'in line' notation) ou de notation déportée (i.e. 'stand off' notation) est utilisé. Le critère de choix entre les deux est un compromis entre la puissance d'expression et la lisibilité. Une notation déportée est plus puissante mais moins lisible.

Tout élément référencé doit apparaître avant son référenceur pour éviter aux développeurs qui utilisent des dispositifs d'analyse XML à la volée comme SAX (Simple API for XML) ou StAX (Streaming API for XML) d'avoir à gérer des références en avant dans le flux XML [6]. Evidemment, pour les développeurs qui utilisent des dispositifs d'analyse en mode DOM (Document Object Model) cette contrainte n'offre aucun intérêt mais ne les empêche pas d'analyser le fichier XML.

Le format respecte les deux spécifications ISO en cours d'élaboration qui sont MAF (ISO 24611) et SynAF (ISO 24615)[1]. Comme toutes les spécifications en cours d'élaboration au sein de l'ISO-TC37/SC4, les structures définies doivent être décorées par des catégories de données définies dans le registre de l'ISO (i.e. data category registry, i.e. DCR) [2].

3 Les niveaux d'annotation

Les niveaux sont décrits du bas vers le haut.

3.1 Niveau Token

Un token est une chaîne de caractères minimale. Elle est minimale dans le sens où un éventuel découpage en fragments plus petits n'est pas réalisé car il n'offre aucun intérêt. Le token est la plus petite unité adressable via un identifiant. Le choix de l'algorithme de segmentation du texte en tokens n'est pas imposé aux participants, mais un algorithme est conseillé en annexe-A.

Chaque token est décrit par un identificateur, un empan et un contenu. Un empan est un couple position du premier caractère / position après le dernier caractère **dans le texte d'origine**. La position est donnée à partir de zéro en nombre de caractères et non en nombre d'octets[2]. Tous les caractères sont comptés, qu'ils soient des lettres, des espaces ou des fins de ligne. Le contenu de la balise est la chaîne de caractères telle qu'elle apparaît dans le texte d'origine.

Par exemple, à partir du texte d'entrée « **Les chaises** », nous aurons :

```
<T id="t0" start="0" end="3">Les</T>
<T id="t1" start="4" end="11">chaises</T>
```

3.2 Niveau mot forme ou forme fléchie

La forme fléchie est une annotation construite à partir de la notion de token. Une forme fléchie n'est pas toujours équivalente à un token, même si c'est fréquemment le cas. Ainsi, les tokens «aujourd», «'»et «hui» pourront ne former qu'une seule forme fléchie, de même que les tokens «afin» et «de». Inversement, un même token peut être référencé par différentes formes fléchies.

1 Ce sont des spécifications qui ne peuvent être désignées comme standard ISO dans la mesure où leur statut n'est pas "Published Standard" mais "Committee Draft".

2 Rappelons que si du temps de l'ISO Latin, les notions de caractères et d'octets étaient confondues, ce n'est plus le cas dans le contexte actuel qui suit Unicode, voir par exemple [5]. En Unicode, un caractère est représenté à l'aide d'un ou plusieurs octets. Un octet fait toujours huit bits.

La forme fléchie est un niveau intermédiaire entre le token et le groupe. Alors que le Token est défini par un algorithme de segmentation, la forme fléchie est définie par sa présence dans le dictionnaire ou bien sa reconnaissance en tant qu'entité nommée.

Dans le format d'annotation, chaque forme fléchie est décrite par un identifiant et des références strictement consécutives à un ou plusieurs tokens.

Sur le même texte que précédemment, et en supposant que les mots figurent en début de document, nous aurons :

```
<T id="t0" start="0" end="3">Les</T>
<W id="w0" tokens="t0"/>

<T id="t1" start="4" end="11">chaises</T>
<W id="w1" tokens="t1"/>
```

De manière optionnelle, la forme fléchie peut porter la partie du discours, le lemme, la forme fléchie canonique après redressage éventuel, une combinaison de traits morphologiques et un booléen indiquant si le mot est une tête pour le groupe. La combinaison doit être déclarée en en-tête du fichier, voir plus loin, la section sur la combinaison des traits.

Dans ce cas, nous aurons :

```
<T id="t0" start="0" end="3">Les</T>
<W id="w0" tokens="t0" pos="definiteArticle" lemma="le" form="les" mstag="nP"/>

<T id="t1" start="4" end="11">chaises</T>
<W id="w1" tokens="t1" pos="commonNoun" lemma="chaise" form="chaises" mstag="nPgF" head="true"/>
```

Les unités multi-mots sont représentées par un ensemble de tokens référencés par une unique forme fléchie. Par exemple, pour "aujourd'hui", nous aurons :

```
<T id="t0" start="0" end="7">aujourd</T>
<T id="t1" start="7" end="8">'</T>
<T id="t2" start="8" end="11">hui</T>
<W id="w0" tokens="t0 t1 t2" pos="adverb" lemma="aujourd'hui" form="aujourd'hui"/>
```

Le dispositif permet de représenter une chaîne de caractères qui donne lieu à un redressage orthographique par éclatement. Un même token est alors référencé par plusieurs formes fléchies. Ainsi, en imaginant que, par erreur, la chaîne d'entrée soit: "unetable", il est possible de représenter la décomposition suivante :

```
<T id="t0" start="0" end="8">unetable</T>
<W id="w0" tokens="t0" pos="indefiniteArticle" lemma="un" form="une"/>
<W id="w1" tokens="t0" pos="commonNoun" lemma="table" form="table"/>
```

Il est possible de représenter les agglutinés comme "au" par "à" et "le" de la manière suivante :

```
<T id="t0" start="0" end="2">au</T>
<W id="w0" tokens="t0" pos="preposition" lemma="à" form="à"/>
<W id="w1" tokens="t0" pos="indefiniteArticle" lemma="le" form="le"/>
```

Les références aux tokens doivent être strictement consécutives. Cela signifie qu'il n'est pas possible qu'un mot adresse un token qui se trouve entre deux tokens d'un autre mot. En d'autres termes, il n'est pas possible de représenter un "composé à trou" comme dans le **contre-exemple** suivant :

```
<!--Rappel: ceci est interdit-->
<T id="t0" start="0" end="2">ne</T>
<T id="t1" start="3" end="11">vraiment</T>
<T id="t2" start="12" end="15">pas</T>
<W id="w0" tokens="t0 t2"/>
<W id="w1" tokens="t1"/>
<!--Rappel: ceci est interdit-->
```

Concernant la liste des parties du discours, le registre de l'ISO contient de multiples valeurs, par exemple "bullet" ou "comma" qui sont organisées en une ontologie. Nous choisissons de ne prendre qu'une liste relativement simple et plate.

Les parties du discours sont les suivantes :

identifiant ISO	nom français
adverb	adverbe
commonNoun	nom commun
coordinatingConjunction	conjonction de coordination
definiteArticle	article défini
demonstrativeDeterminer	déterminant démonstratif
exclamativeDeterminer	déterminant exclamatif
foreignText	texte étranger
indefiniteDeterminer	déterminant indéfini
interjection	interjection
letter	lettre
mainPunctuation	ponctuation principale
negativeParticle	particule négative
numeral	numéral
ordinalAdjective	adjectif ordinal
personnalPronoun	pronom personnel
possessiveDeterminer	déterminant possessif
possessivePronoun	pronom possessif
preposition	préposition
properNoun	nom propre
qualifierAdjective	adjectif qualificatif
relativePronoun	pronom relatif
residual	résiduel
secondaryPunctuation	ponctuation secondaire
subordinatingConjunction	conjonction de subordination
verb	verbe

3.3 Niveau Groupe

Il s'agit de représenter un groupe syntaxique. Ce groupe peut être un groupe non récursif ou bien un syntagme récursif.

Un groupe est constitué d'un identifiant, d'un type syntaxique, d'une suite de formes fléchies ou de groupes. Un groupe possède au moins une forme fléchie ou un groupe, autrement dit, un groupe ne peut pas être vide.

```
<T id="t0" start="0" end="3">Les</T>
<T id="t1" start="4" end="11">chaises</T>
<G id="g0" type="GN">
        <W id="w0" tokens="t0"/>
        <W id="w1" tokens="t1"/>
</G>
```

L'attribut 'type' est valué par une des constante de la liste donnée ci-dessous. Chaque valeur trouve un correspondant dans le registre de catégorie de données de l'ISO (DCR). Dans le tableau qui suit, la colonne de gauche est la valeur utilisée dans le projet Easy/Passage, la colonne du milieu, le nom français dans le DCR et la colonne de droite est l'identifiant dans le DCR.

nom Easy	nom ISO français	identifiant ISO
NV	noyau verbal	verbNucleus
GN	groupe nominal	nounPhrase
GP	groupe prépositionnel	prepositionPhrase
GA	groupe adjectival	adjectivePhrase
GR	groupe adverbial	adverbPhrase
PV	groupe verbal prépositionnel	prepositionVerbPhrase

De plus, les valeurs GV (pour groupe verbal), GD (pour groupe déterminant) et CL (pour clause, ou proposition) peuvent être utilisées. Notons que ces valeurs n'existaient pas dans la campagne Easy puisque cette dernière ne permettait que l'annotation en chunks (i.e. groupes non-récursifs) sans aucune possibilité pour un groupe d'avoir des sous-groupes.

De manière optionnelle, le groupe peut porter une combinaison de traits morpho-syntaxiques comme par exemple :

```
<T id="t0" start="0" end="3">Les</T>
<T id="t1" start="4" end="11">chaises</T>
<G id="g0" type="GN" mstag="nP gF">
        <W id="w0" tokens="t0"/>
        <W id="w1" tokens="t1"/>
</G>
```

Notons que lors de la campagne Technolangue/Easy, les groupes étaient enchâssés dans une balise "groupes". Mais lors de la première campagne de Passage, cette balise a disparu, nous avons fait de même.

3.4 Niveau Relation

C'est une notation déportée (i.e. 'stand-off'). Pour des raisons de compatibilité avec les outils de la campagne Easy, le format général des relations est conservé.

Les arguments référencés sont aussi bien des formes fléchies que des groupes, mais au contraire du format Easy, les références sont de véritables références XML. En effet, dans le format Easy, les références avaient été déclarés par erreur en type CDATA optionnel, ce qui faisait que les outils XML ne pouvaient pas vérifier le typage des éléments.

Par rapport à la DTD Easy, les éléments "adjectif" et "complementeur" ont été ajoutés car ils étaient mentionnés dans le contenu de l'élément "relation" sans être définis. Pour détecter automatiquement ce type de problème, un validateur XML[3] a été utilisé.

En supposant que l'on veuille définir une relation sujet entre le groupe g0 et g1, nous aurons la structure suivante :

3 Validome en l'occurrence, disponible en tant que service web gratuit à http://www.validome.org/xml

```
<R id=”r0" type="SUJ_V">
        <sujet  ref="g0"/>
        <verbe ref="g1"/>
</R>
```

L’attribut ‘type’ est valué par une des constantes de la liste donnée ci-dessous. Chaque valeur trouve un correspondant dans le registre de catégorie de données de l’ISO (DCR). Dans le tableau qui suit, la colonne de gauche est la valeur utilisée dans le projet Easy/Passage, la colonne du milieu, le nom français dans le DCR et la colonne de droite est l’identifiant dans le DCR.

nom Easy	nom ISO français	identifiant ISO
SUJ_V	sujet	subject
AUX_V	auxiliaire	auxiliary
COD_V	objet direct	directObject
CPL_V	complément verbal	verbComplement
MOD_V	modifieur de verbe	verbModifier
COMP	complémenteur	complementizer
ATB_SO	attribut	attribute
MOD_N	modifieur de nom	nounModifier
MOD_A	modifieur d'adjectif	adjectiveModifier
MOD_R	modifieur d'adverbe	adverbModifier
MOD_P	modifieur de préposition	prepositionModifier
COORD	coordination	coordination
APP	apposition	apposition
JUXT	juxtaposition	juxtaposition

Le rôle est indiqué par l'une des valeurs figurant ci-dessous :

nom Easy	nom ISO français	identifiant ISO
adjectif	adjectif	adjective
adverbe	adverbe	adverb
appose	apposé	apposed
attribut	attribut	attribute
auxiliaire	auxiliaire	auxiliary
cod	objet direct	directObject
complement	complément verbal	verbalComplement
complementeur	complémenteur	complementizer
coord-d	coordonné droit	rightCoordinated
coord-g	coordonné gauche	leftCoordinated
coordonnant	coordonnant	coordinator
modifieur	modifieur	modifier
nom	nom	noun
premier	premier	first
preposition	préposition	preposition
suivant	suivant	next
sujet	sujet	subject
verbe	verbe	verb

3.5 La pose de repères

Un repère est un dispositif qui permet à un annotateur humain d'enregistrer un commentaire qui lui permettra de localiser un fragment du texte afin d'y revenir par la suite. Un repère peut aussi être enregistré automatiquement par un programme. Un repère référence soit un empan, soit un ou plusieurs éléments du fichier via les identifiants.

Nous aurons par exemple :

```
<M id=”m0” start="0" end="3" objs="">à valider avec Robert<M/>
```

3.6 Les entités nommées

En terme de format, une entité nommée est une annotation qui possède a minima un identifiant, un type et une liste de références de mot-formes. Ces références ne sont pas nécessairement continues mais leur portée ne doit pas excéder la phrase. De manière optionnelle, l'entité nommée possède un sous-type et une combinaison de traits morpho-syntaxiques.

Les types possibles sont les suivants:

identifiant	exemple
individual	Jacques Chirac
organization	Danone
location	Dijon
dateTime	mardi 5 février
URLetc	www.afnor.org
measure	2,66 GHz
mark	PDF

Notons que la détermination d'une liste de types d'entités nommées est une entreprise délicate. Le niveau de détail est très variable d'un logiciel à l'autre. Et, contrairement aux autres listes de valeurs, il n'existe pas de liste fixée par l'ISO: il y a bien un travail en cours pour la représentation des entités nommées mais ce dernier n'est pas assez avancé pour fixer une liste de valeurs. Le choix est pris de fixer une liste de niveau relativement général, sur lequel tout le monde semble à peu près d'accord et de laisser le sous-type complètement libre.

L'entité nommée n'est pas un élément inclus dans un mot ou dans un groupe. Il n'est pas dans le 'flux' mais figure en quelque sorte de manière 'parallèle' aux groupes. Il est un élément d'annotation de 'première classe'.

Notons que la définition linguistique de ce qu'est une entité nommée et quelles sont les consignes pour la construire ne fait pas l'objet du présent document mais doit être défini par un guide d'annotation.

Si nous avons coïncidence entre le groupe et l'entité nommée, nous aurons sur "Jacques Chirac" :

```
<T id="t0" start="0" end="7">Jacques</T>
<T id="t1" start="8" end="13">Chirac</T>
<G id="g0" type="GN">
        <W id="w0" tokens="t0"/>
        <W id="w1" tokens="t1"/>
</G>
<NE id="e0" type="individual" lst="w0 w1"/>
```

Mais nous pouvons ne pas avoir une coïncidence exacte entre le groupe et l'entité nommée. En supposant que le guide d'annotation spécifie que l'article ne fait pas partie de l'entité nommée, nous aurons une entité nommée plus petite que le groupe. Ainsi sur "l'Hôtel Crillon" par exemple, nous aurons :

```
<T id="t0" start="0" end="2">l'</T>
<T id="t1" start="2" end="7">Hôtel</T>
<T id="t2" start="8" end="15">Crillon</T>
<G id="g0" type="GN">
        <W id="w0" tokens="t0"/>
        <W id="w1" tokens="t1"/>
        <W id="w2" tokens="t2"/>
</G>
<NE id="e0" type="location" lst="w1 w2"/>
```

Inversement, le format d'annotation permet de construire un empan plus grand que le groupe avec, par exemple sur "le jour de la Toussaint" avec l'entité nommée « jour de la Toussaint »:

```
<T id="t0" start="0" end="2">le</T>
<T id="t1" start="3" end="7">jour</T>
<T id="t2" start="8" end="10">de</T>
<T id="t3" start="11" end="13">la</T>
<T id="t4" start="14" end="23">Toussaint</T>
<G id="g0" type="GN">
        <W id="w0" tokens="t0"/>
        <W id="w1" tokens="t1"/>
</G>
<G id="g1" type="GP">
        <W id="w2" tokens="t2"/>
        <W id="w3" tokens="t3"/>
        <W id="w4" tokens="t4"/>
</G>
<NE id="e0" type="dateTime" lst="w1 w2 w3 w4"/>
```

Le format permet aussi de gérer des coordinations plus complexes si nécessaire. Par exemple, en supposant l'entrée: "Bill et Hillary Clinton", il est possible de produire d'un côté "Bill Clinton" et de l'autre "Hillary Clinton". Les références des entités vers les mots-formes ne sont, dans ce cas particulier, alors pas continues.

3.7 La définition des combinaisons de traits morpho-syntaxiques

En début de fichier, de manière optionnelle, les combinaisons des traits morpho-syntaxiques (i.e. tagsets en anglais) pourront être définies.

Une combinaison porte un nom et un ensemble de traits (i.e. features) qui en forme le vocabulaire. Chacun de ces traits doit être une valeur du DCR. La syntaxe est celle de la spécification ISO pour les structures de traits [7].

Nous aurons, par exemple :

```
<MSTAG id="nP">
        <fs>
                <f name="grammaticalNumber">
                        <symbol value="plural"/>
                </f>
        </fs>
</MSTAG>
```

Les traits morpho-syntaxiques pour le français sont les suivants :

catégorie de donnée utilisée comme attribut	domaine conceptuel
grammaticalNumber	singular
	plural
grammaticalGender	masculine
	feminine
person	firstPerson
	secondPerson
	thirdPerson
verbFormMood	indicative
	conditional
	subjunctive
	imperative
	participle
	infinitive
grammaticalTense	present
	imperfect
	past
	future
ownerPerson	firstPerson
	secondPerson
	thirdPerson
ownerNumber	singular
	plural
ownedGender	masculine
	feminine
ownedNumber	singular
	plural

Il est possible de spécifier des disjonctions, par exemple dans les situations où l'on ne désire pas statuer sur un trait précis et conserver l'ambiguïté, comme dans le fragment XML suivant :

```
<MSTAG id="mI.S">
        <fs>
                <f name="grammaticalMood">
                        <vAlt>
                                <symbol value="indicative"/>
                                <symbol value="subjunctive"/>
                        </vAlt>
                </f>
        </fs>
</MSTAG>
```

3.8 Niveau Sentence ou phrase

C'est une notation enchâssée. Le choix de l'enchâssement plutôt que la notation déportée est motivé par l'intérêt de rassembler toutes les balises d'une même phrase dans la même partie du fichier, et ainsi en faciliter la lecture.

Une phrase est composée d'une suite non vide de groupes et de formes fléchies, suivie d'une suite éventuellement vide de relations, suivie d'une suite éventuellement vide de repères.

L'élément "Sentence" possède un attribut optionnel qui s'appelle "trust" qui permet d'indiquer le degré de confiance que le producteur de la donnée accorde à la qualité de l'analyse. La valeur de l'attribut varie de 0 (pas fiable) à 100 (fiable).

Nous aurons par exemple :

```
<Sentence trust="100">
        <G>…</G>
        <G>…</G>
        <W>…</W>
        <G>…</G>
        <W>…</W>
        <R>…</R>
        <R>…</R>
        <M>…</M>
        <M>…</M>
</Sentence>
```

3.9 Niveau Document

Le niveau document se résume à une balise englobante des phrases du texte en question. La balise de document possède l'attribut suivant :

- un attribut qui est le nom de fichier. Ce nom n'est pas le nom avec le chemin d'accès au sens du système d'exploitation, mais au contraire, la forme relative.

```
<Document file="monfichier.xml">
        <Sentence …>…</Sentence>
        <Sentence …>…</Sentence>
</Document>
```

4 Conclusion

Comparativement au format Easy, ce format s'en distingue en trois points importants :

- récursion possible,
- annotation morpho-syntaxique possible,
- segmentation en tokens optionnelle spécifiée.

Ce format est plus puissant que le format du projet Easy, donc, mais pour être correctement utilisé, une mise à jour du guide d'annotation s'avère nécessaire.

Annexe A: Segmentation en tokens

L'algorithme décrit ci-dessous est préconisé mais non obligatoire. C'est cet algorithme qui sera utilisé pour la mise à disposition de la banque de données (i.e. treebank) finale.

L'algorithme est très simple et se fonde sur les propriétés des caractères telles que définies par Unicode dans la table General_Category[4]. Rappelons que les caractères sont définis par une hiérarchie à deux niveaux des propriétés. Le premier niveau est soit l'une des 7 valeurs suivantes : Other, Letter, Mark, Number, Punctuation, Symbol et Separator. Notons de plus que le deuxième niveau est par exemple Lowercase_Letter qui est une valeur plus spécifique de Letter, mais nous n'avons pas besoin de ce deuxième niveau.

Pour ce qui nous concerne, nous distinguons :

- **les caractères informatifs** qui vont constituer l'essentiel du contenu textuel. Ce sont les caractères de type Letter et Number.

- **les caractères séparateurs**. Ce sont les caractères de type Separator qui comprennent par exemple, l'espace ou le saut de ligne.

- **les caractères autonomes**. Ce sont les caractères de type Other, Mark, Punctuation, Symbol. La virgule par exemple sera de type autonome.

La segmentation en tokens est régie par les principes suivants:

Principe#1

Un token est défini de manière exclusive :

- soit comme une suite contiguë de caractères de type Letter ou Number, à l'exclusion de tout autre caractère. Par exemple, les suites "la", "34", "X56" formeront chacun un token.

- soit comme un caractère autonome. Par exemple, la virgule formera un token à elle toute seule, et ceci quels que soient les caractères qui l'entourent.

Principe#2

Un caractère séparateur ne figure pas dans l'annotation. Par exemple, le caractère espace même s'il provoque une segmentation, ne figure pas dans le résultat de l'annotation.

Notons que la conséquence de ces deux principes est qu'un token ne peut pas être vide.

4 Voir le site www.unicode.org ou bien la présentation un peu plus compréhensible de [Gesgraupes] : table des propriétés, page 200.

Annexe B: DTD XML du format Passage

```
<?xml version='1.0' encoding="UTF-8"?>
<!ELEMENT Document (MSTAG*, Sentence*)>
<!ATTLIST Document      dtdVersion      CDATA #FIXED "1.1"
                        file            CDATA #IMPLIED>
<!-- ================================-->
<!ELEMENT MSTAG (fs)>
<!ATTLIST MSTAG         id              ID #REQUIRED>
<!ELEMENT fs (f+)>
<!ELEMENT f (symbol | vAlt)+>
<!ATTLIST f             name            CDATA #REQUIRED>
<!ELEMENT symbol EMPTY>
<!ATTLIST symbol        value           CDATA #REQUIRED>
<!ELEMENT vAlt (symbol+)>
<!-- ================================-->
<!ELEMENT Sentence ( (T|G|W)+, R*, M* ,NE*)>
<!ATTLIST Sentence      trust           CDATA #IMPLIED>
<!-- ================================-->
<!ELEMENT T (#PCDATA)>
<!ATTLIST T             id              ID #REQUIRED
                        start           CDATA #REQUIRED
                        end             CDATA #REQUIRED>
<!-- ================================-->
<!ELEMENT G (W+)>
<!ATTLIST G             id              ID #REQUIRED
                        type            (NV|GN|GP|GA|GR|PV)  #REQUIRED
                        mstag           IDREFS #IMPLIED>
<!-- ================================-->
<!ELEMENT W EMPTY>
<!ATTLIST W             id              ID #REQUIRED
                        tokens          IDREFS #REQUIRED
                        pos             (adverb | commonNoun |
                                        coordinatingConjunction | definiteArticle |
                                        demonstrativeDeterminer | exclamativeDeterminer |
                                        foreignText | indefiniteDeterminer|interjection |
                                        letter | mainPunctuation |
                                        negativeParticle|numeral | ordinalAdjective |
                                        personnalPronoun | possessiveDeterminer |
                                        possessivePronoun | preposition|properNoun |
                                        qualifierAdjective | relativePronoun | residual |
                                        secondaryPunctuation | subordinatingConjunction |
                                        verb) #IMPLIED
                        lemma           CDATA #IMPLIED
                        form            CDATA #IMPLIED
                        mstag           IDREFS #IMPLIED
                         head           (true | false) #IMPLIED>
<!-- ================================-->
<!ELEMENT R (adjectif | adverbe | appose | attribut | auxiliaire | cod | complement | complementeur |
coord-d | coord-g | coordonnant | modifieur | nom | premier | preposition | s-o | suivant | sujet |
verbe )+ >
<!ATTLIST R             id              ID #REQUIRED
                        type            (APPOS|ATB-SO|AUX-V|COMP|COD-V|COORD|
CPL-V|JUXT|MOD-A|MOD-N|MOD-P|MOD-R|MOD-V|SUJ-V) #REQUIRED >

<!ELEMENT adjectif EMPTY >
<!ATTLIST adjectif      ref             IDREF #REQUIRED>

<!ELEMENT adverbe EMPTY >
<!ATTLIST adverbe       ref             IDREF #REQUIRED>

<!ELEMENT appose EMPTY >
<!ATTLIST appose        ref             IDREF #REQUIRED>
```

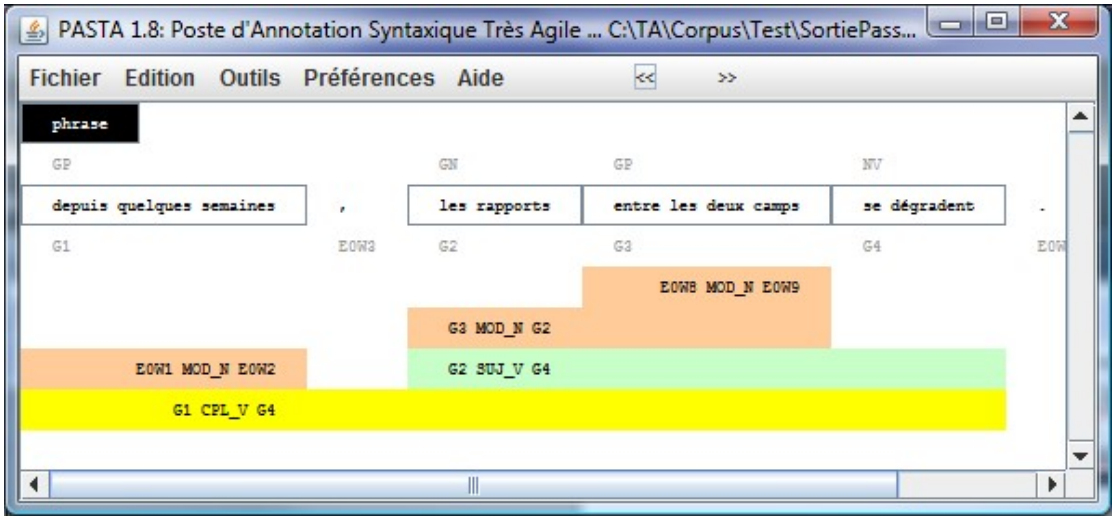
```
<!ELEMENT attribut EMPTY >
<!ATTLIST attribut          ref                 IDREF #REQUIRED>

<!ELEMENT auxiliaire EMPTY >
<!ATTLIST auxiliaire        ref                 IDREF #REQUIRED>

<!ELEMENT cod EMPTY >
<!ATTLIST cod               ref                 IDREF #REQUIRED>

<!ELEMENT complement EMPTY >
<!ATTLIST complement        ref                 IDREF #REQUIRED>

<!ELEMENT complementeur EMPTY >
<!ATTLIST complementeur           ref           IDREF #REQUIRED>

<!ELEMENT coord-d EMPTY >
<!ATTLIST  coord-d          ref                 IDREF #REQUIRED>

<!ELEMENT coord-g EMPTY >
<!ATTLIST  coord-g          ref                 IDREF #IMPLIED>

<!ELEMENT coordonnant EMPTY >
<!ATTLIST  coordonnant      ref                 IDREF #REQUIRED>

<!ELEMENT modifieur EMPTY >
<!ATTLIST  modifieur        ref                 IDREF #REQUIRED>

<!ELEMENT nom EMPTY >
<!ATTLIST  nom                    ref                   IDREF #REQUIRED>

<!ELEMENT premier EMPTY >
<!ATTLIST  premier          ref                 IDREF #REQUIRED>

<!ELEMENT preposition EMPTY >
<!ATTLIST  preposition      ref                 IDREF #REQUIRED>

<!ELEMENT s-o EMPTY >
<!ATTLIST s-o    valeur ( sujet | objet | ind ) #REQUIRED >

<!ELEMENT suivant EMPTY >
<!ATTLIST  suivant          ref                 IDREF #REQUIRED>

<!ELEMENT sujet EMPTY >
<!ATTLIST  sujet                  ref                   IDREF #REQUIRED>

<!ELEMENT verbe EMPTY >
<!ATTLIST  verbe            ref                 IDREF #REQUIRED>
<!-- ================================-->
<!ELEMENT M (#PCDATA)>
<!ATTLIST M                 id                  ID #REQUIRED
                            start               CDATA #IMPLIED
                            end                 CDATA #IMPLIED
                            objs                IDREFS #IMPLIED>
<!-- ================================-->
<!ELEMENT NE EMPTY>
<!ATTLIST NE                id                  ID #REQUIRED
                            type                (individual | organization | location | dateTime |
                                                URLetc | measure | mark) #REQUIRED
                            subType             CDATA #IMPLIED
                            lst                 IDREFS #REQUIRED
                            mstag               IDREFS #IMPLIED>
```

Annexe C: Exemples

Exemple#1

Sur la phrase suivante, située en début de fichier:

« Depuis quelques semaines, les rapports entre les deux camps se dégradent. »



Nous aurons le balisage XML suivant:

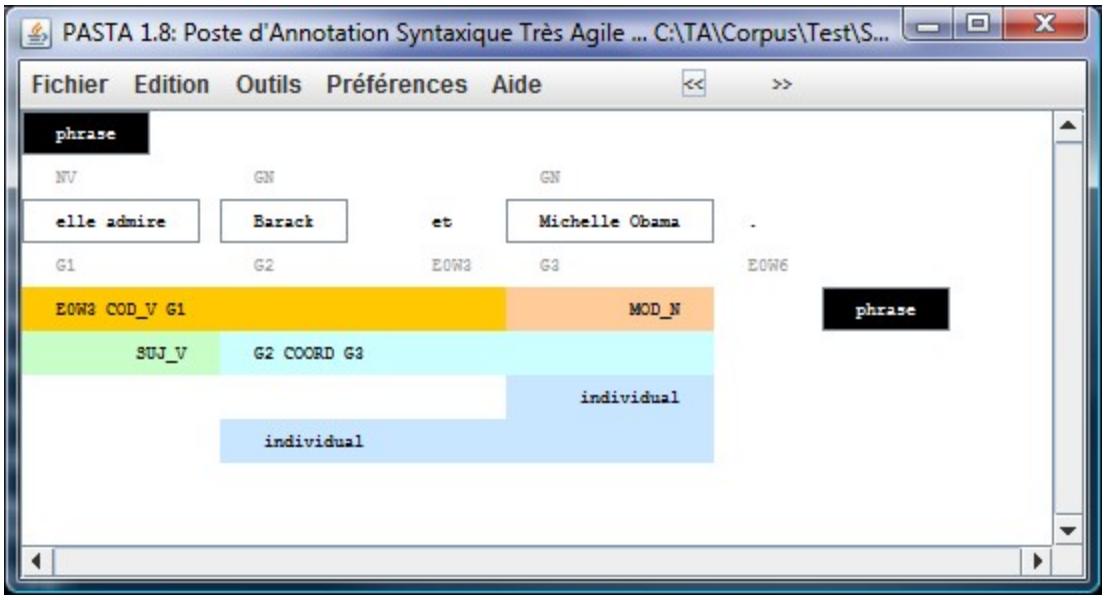
```
<?xml version="1.0" encoding="UTF-8"?>
<Document dtdVersion="1.1" file="test.txt">
<MSTAG id="nS"><fs><f name="grammaticalNumber"><symbol value="singular"/></f></fs></MSTAG>
<MSTAG id="nP"><fs><f name="grammaticalNumber"><symbol value="plural"/></f></fs></MSTAG>
<MSTAG id="nX"><fs><f name="grammaticalNumber"><vAlt><symbol value="singular"/><symbol value="plural"/></vAlt></f></fs></MSTAG>
<MSTAG id="gM"><fs><f name="grammaticalGender"><symbol value="masculine"/></f></fs></MSTAG>
<MSTAG id="gF"><fs><f name="grammaticalGender"><symbol value="feminine"/></f></fs></MSTAG>
<MSTAG id="gE"><fs><f name="grammaticalGender"><vAlt><symbol value="masculine"/><symbol value="feminine"/></vAlt></f></fs></MSTAG>
<MSTAG id="mI"><fs><f name="verbFormMood"><symbol value="infinitive"/></f></fs></MSTAG>
<MSTAG id="mP"><fs><f name="verbFormMood"><symbol value="participle"/></f></fs></MSTAG>
<MSTAG id="mC"><fs><f name="verbFormMood"><symbol value="conjugated"/></f></fs></MSTAG>
<MSTAG id="mU"><fs><f name="verbFormMood"><symbol value="unknown"/></f></fs></MSTAG>
<Sentence>
   <T id="E0W0T0" start="0" end="6">depuis</T>
   <T id="E0W1T0" start="7" end="15">quelques</T>
   <T id="E0W2T0" start="16" end="24">semaines</T>
 <G id="E0G1" type="GP">
   <W id="E0W0" tokens="E0W0T0" pos="preposition" lemma="depuis" form="depuis"/>
   <W id="E0W1" tokens="E0W1T0" pos="indefiniteArticle" lemma="quelque" form="quelques"/>
   <W id="E0W2" tokens="E0W2T0" pos="commonNoun" lemma="semaine" form="semaines" mstag="nP gF"/>
 </G>
   <T id="E0W3T0" start="24" end="25">,</T>
   <W id="E0W3" tokens="E0W3T0" pos="secondaryPunctuation" lemma="," form=","/>
   <T id="E0W4T0" start="26" end="29">les</T>
   <T id="E0W5T0" start="30" end="38">rapports</T>
 <G id="E0G2" type="GN">
   <W id="E0W4" tokens="E0W4T0" pos="definiteArticle" lemma="le" form="les"/>
   <W id="E0W5" tokens="E0W5T0" pos="commonNoun" lemma="rapport" form="rapports" mstag="nP gM"/>
 </G>
   <T id="E0W6T0" start="39" end="44">entre</T>
   <T id="E0W7T0" start="45" end="48">les</T>
   <T id="E0W8T0" start="49" end="53">deux</T>
   <T id="E0W9T0" start="54" end="59">camps</T>
 <G id="E0G3" type="GP">
   <W id="E0W6" tokens="E0W6T0" pos="preposition" lemma="entre" form="entre"/>
   <W id="E0W7" tokens="E0W7T0" pos="definiteArticle" lemma="le" form="les"/>
   <W id="E0W8" tokens="E0W8T0" pos="numeral" lemma="deux" form="deux"/>
   <W id="E0W9" tokens="E0W9T0" pos="commonNoun" lemma="camp" form="camps" mstag="nP gM"/>
 </G>
   <T id="E0W10T0" start="60" end="62">se</T>
   <T id="E0W11T0" start="63" end="72">dégradent</T>
 <G id="E0G4" type="NV">
   <W id="E0W10" tokens="E0W10T0" pos="weakPersonalPronoun" lemma="se" form="se"/>
   <W id="E0W11" tokens="E0W11T0" pos="verb" lemma="dégrader" form="dégradent" mstag="mC"/>
 </G>
   <T id="E0W12T0" start="72" end="73">.</T>
```

```
  <W id="E0W12" tokens="E0W12T0" pos="declarativePunctuation" lemma="." form="."/>
 <R id="E0R0" type="SUJ-V">
  <sujet ref="E0G2"/>
  <verbe ref="E0G4"/>
 </R>
 <R id="E0R1" type="CPL-V">
  <complement ref="E0G1"/>
  <verbe ref="E0G4"/>
 </R>
 <R id="E0R2" type="MOD-N">
  <modifieur ref="E0W1"/>
  <nom ref="E0W2"/>
 </R>
 <R id="E0R3" type="MOD-N">
  <modifieur ref="E0G3"/>
  <nom ref="E0G2"/>
 </R>
 <R id="E0R4" type="MOD-N">
  <modifieur ref="E0W8"/>
  <nom ref="E0W9"/>
 </R>
 </Sentence>
```

Exemple#2

Sur la phrase suivante, située en début de fichier:

« Elle admire Barack et Michelle Obama. »



Nous aurons le balisage XML suivant, avec deux entités nommées « Barack Obama » et : « Michelle Obama » en fin de fichier.

```
<?xml version="1.0" encoding="UTF-8"?>
<Document dtdVersion="1.1" file="test.txt">
<MSTAG id="nS"><fs><f name="grammaticalNumber"><symbol value="singular"/></f></fs></MSTAG>
<MSTAG id="nP"><fs><f name="grammaticalNumber"><symbol value="plural"/></f></fs></MSTAG>
<MSTAG id="nX"><fs><f name="grammaticalNumber"><vAlt><symbol value="singular"/><symbol value="plural"/></vAlt></f></fs></MSTAG>
<MSTAG id="gM"><fs><f name="grammaticalGender"><symbol value="masculine"/></f></fs></MSTAG>
<MSTAG id="gF"><fs><f name="grammaticalGender"><symbol value="feminine"/></f></fs></MSTAG>
<MSTAG id="gE"><fs><f name="grammaticalGender"><vAlt><symbol value="masculine"/><symbol value="feminine"/></vAlt></f></fs></MSTAG>
<MSTAG id="mI"><fs><f name="verbFormMood"><symbol value="infinitive"/></f></fs></MSTAG>
<MSTAG id="mP"><fs><f name="verbFormMood"><symbol value="participle"/></f></fs></MSTAG>
<MSTAG id="mC"><fs><f name="verbFormMood"><symbol value="conjugated"/></f></fs></MSTAG>
<MSTAG id="mU"><fs><f name="verbFormMood"><symbol value="unknown"/></f></fs></MSTAG>
<Sentence>
   <T id="E0W0T0" start="0" end="4">elle</T>
   <T id="E0W1T0" start="5" end="11">admire</T>
 <G id="E0G1" type="NV">
   <W id="E0W0" tokens="E0W0T0" pos="weakPersonalPronoun" lemma="il" form="elle" mstag="nS gM"/>
   <W id="E0W1" tokens="E0W1T0" pos="verb" lemma="admirer" form="admire" mstag="mC"/>
 </G>
   <T id="E0W2T0" start="12" end="18">Barack</T>
 <G id="E0G2" type="GN">
   <W id="E0W2" tokens="E0W2T0" pos="properNoun" lemma="Barack" form="Barack" mstag="gM"/>
 </G>
   <T id="E0W3T0" start="19" end="21">et</T>
   <W id="E0W3" tokens="E0W3T0" pos="coordinatingConjunction" lemma="et" form="et"/>
   <T id="E0W4T0" start="22" end="30">Michelle</T>
   <T id="E0W5T0" start="31" end="36">Obama</T>
```

```
  <G id="E0G3" type="GN">
    <W id="E0W4" tokens="E0W4T0" pos="properNoun" lemma="Michelle" form="Michelle" mstag="gF"/>
    <W id="E0W5" tokens="E0W5T0" pos="properNoun" lemma="Obama" form="Obama"/>
  </G>
    <T id="E0W6T0" start="36" end="37">.</T>
    <W id="E0W6" tokens="E0W6T0" pos="declarativePunctuation" lemma="." form="."/>
  <R id="E0R0" type="SUJ-V">
    <sujet ref="E0W0"/>
    <verbe ref="E0W1"/>
  </R>
  <R id="E0R1" type="COD-V">
    <cod ref="E0W3"/>
    <verbe ref="E0G1"/>
  </R>
  <R id="E0R2" type="MOD-N">
    <modifieur ref="E0W4"/>
    <nom ref="E0W5"/>
  </R>
  <R id="E0R3" type="COORD">
    <coordonnant ref="E0W3"/>
    <coord-g ref="E0G2"/>
    <coord-d ref="E0G3"/>
  </R>
  <NE id="E0E0" type="individual" lst="E0W2 E0W5"/>
  <NE id="E0E1" type="individual" lst="E0W4 E0W5"/>
</Sentence>
</Document>
```

Annexe D: Références

[1] Rumbaugh J., Jacobson I., Booch G. 2005 The Unified Modeling language reference manual, 2nd Edition, Addison Wesley, Boston MA

[2] Registre de catégories de données http://syntax.inist.fr

[3] MAF, voir http://lirics.loria.fr/doc_pub/maf.pdf

[4] SynAF, voir http://lirics.loria.fr/doc_pub/N421_SynAF_CD_ISO_24615.pdf

[5] Desgraupes B. 2005 Passeport pour Unicode, Vuibert, Paris

[6] Harold & Means 2004 XML in a nutshell 3rd Edition, O'Reilly, Sebastopol, CA

[7] Language Resources Management - Feature structures - Part 1: feature structure representation ISO 24610-1