# Constitution de banques de textes multilingues : un mécanisme fondé sur le standard XML

*Nous présentons dans cet article une méthodologie pour la réalisation de ressources linguistiques réutilisables, à savoir des documents munis de balises, en vue de leur intégration dans des banques de données textuelles. Après avoir présenté les mérites du standard de balisage XML, nous proposons l'utilisation de la spécification XCES, application de XML, pour le balisage des ressources documentaires, et nous la décrivons brièvement. Enfin, nous traçons les principales étapes du processus de conversion des ressources vers ce format, et suggérons quelques repères pédagogiques pouvant aider les chercheurs qui abordent ces tâches de conversion.*

*Termes-clés :*
*banques de textes ; balisage de textes ; XML ; XCES ; conversion des ressources.*

## 1 Introduction : motivation et objectifs

L'UTILITÉ des banques de textes dans le domaine du traitement informatique des documents, dans le contexte de la société d'information, est désormais établie. Les banques de textes sont utilisées, par exemple, pour la construction de ressources lexicographiques et terminologiques, pour les études linguistiques, en particulier de corpus, et pour la mise au point d'outils de traitement de la langue. Les banques de textes multilingues fournissent un matériel appréciable aux études comparatives et, lorsque les textes dans plusieurs langues sont alignés avec une précision raisonnable, ces banques permettent la construction de lexiques multilingues et de ressources pour les outils d'aide à la traduction, tels les mémoires de traduction. Toutefois, pour que les banques de textes puissent jouer ce rôle, il est nécessaire qu'elles soient convenablement *structurées*. En d'autres mots, le contenu des banques doit être décrit de façon interprétable et uniforme, et, dans la mesure du possible, la structure des textes qui constituent la banque doit être également mise en évidence. C'est donc la structuration dans un format explicite et transparent qui distingue une banque de textes d'une simple collection de documents.

Dans cet article, nous nous proposons de décrire une méthode, des outils et un format permettant la structuration des banques de textes, à partir de ressources documentaires préexistantes, dans différentes langues. Le fondement en est le standard XML pour le marquage des documents, sur lequel se fonde le standard d'annotation de corpus XCES, que nous proposons d'utiliser. Ce dernier conçoit les documents comme formés d'un en-tête, contenant l'information *à propos du* document, et d'un corps, contenant le document lui-même, tous deux structurés selon des spécifications précises. Une procédure pour l'application de ces standards, ainsi que des éléments pédagogiques, sera également décrite.

Le cadre proposé ici, développé en liaison avec le programme de formation du Rifal pour 2002-2003, est tout particulièrement destiné aux institutions et aux chercheurs qui souhaitent augmenter le potentiel d'une collection de documents électroniques disponibles, en particulier dans des langues encore peu représentées dans la société de l'information, par exemple sur Internet. En effet, un des objectifs du Rifal est d'aider ses membres du Sud à se doter de banques de données terminologiques et textuelles multilingues capables de répondre aux besoins locaux tout en constituant une ressource globalement distribuée. Alors que les formations antérieures du Rifal se sont concentrées sur les techniques terminologiques et la conversion des données existantes, les formations à venir porteront sur la construction de banques de données textuelles et terminologiques multilingues. Le logiciel BTML (Yang 2001), paramétré selon les besoins du Rifal, permettra la gestion de ces banques multilingues distribuées, accessibles *via* Internet.

## 2 Notions théoriques

Nous présentons en premier lieu dans cette section les préliminaires théoriques aux méthodes que nous proposons. Il s'agit d'abord du standard XML, qui définit une syntaxe abstraite pour le marquage de documents à l'aide de balises ainsi que des outils associés, tels les DTD (définition du type de document) et le langage XSLT (langage extensible pour les feuilles de style). Puis nous décrirons brièvement et montrerons l'intérêt des spécifications XCES (standard pour l'encodage des corpus en XML) par rapport à nos objectifs[1].

1 Nous employons le mot « standard » pour désigner ces spécifications, bien qu'aucun organisme normatif ne les ait adoptées jusqu'à présent (sauf pour SGML, *cf.* section 2.1.1). Il s'agit plutôt de standards auxquels la communauté choisit de se conformer pour des raisons de compatibilité et d'échange.

## 2.1 Le standard de marquage XML et les outils associés

### 2.1.1 Origines et intérêt

Le langage XML (en anglais, *eXtended Markup Language*, ou langage de marquage étendu) permet de marquer les documents grâce à des *balises*, afin de les structurer. L'idée centrale est de marquer le contenu d'un document plutôt que sa forme, par exemple le sens de ses différentes parties. Cette façon de rendre explicite le contenu des documents est flexible et expressive, ce qui fait que les applications d'XML sont très nombreuses. Les documents structurés en XML s'échangent facilement entre les humains ou entre des programmes de traitement conçus pour utiliser tel ou tel jeu de balises. XML est très utilisé pour les applications multilingues, grâce aussi à ses liens avec Unicode, jeu de caractères universel (*cf.* section 3.2 ci-dessous) : un document XML est un simple document texte, muni de balises et écrit avec un jeu de caractères précis, par défaut Unicode/UTF-8. Outre les premières lignes qui déclarent le document comme étant écrit « en XML », un document XML contient du texte, enchâssé dans une série hiérarchique de balises. Par ailleurs, XML, standard libre de droits pour la définition des balises, ne doit pas être confondu avec un langage de programmation qui définit des instructions.

Signalons, d'un point de vue historique, que l'ancêtre de XML et de HTML est le langage SGML (*Standard Generalized Markup Language*, ou langage général standard pour le marquage), conçu au début des années 1980, et devenu le standard ISO 8879 en 1986. HTML, le langage de formatage de la plupart des documents accessibles sur Internet, fut à l'origine conçu comme une application de SGML en vue de l'affichage, à savoir un jeu de balises particulières respectant la syntaxe SGML. Quant à lui, XML dérive d'une initiative visant à alléger SGML, mais garde toute sa généralité. La spécification XML 1.0, parue en 1998, est en vigueur encore aujourd'hui (*cf.* www.w3.org).

XML présente quelques points communs mais aussi d'importantes différences par rapport à HTML. Ce dernier ne permet qu'un balisage « graphique » des textes, en vue de l'affichage, alors que XML permet de définir des jeux de balises plus riches, plus nuancés, permettant d'exprimer des informations plus détaillées sur la structure et la provenance des textes. Dans XML, les noms et la syntaxe des balises sont à définir selon les besoins, alors qu'en HTML ils sont fixés une fois pour toutes. Enfin, XHTML est une application de XML très proche de HTML, appelée à le remplacer.

### 2.1.2 La DTD, une « grammaire des balises »

Un langage à part permet de définir les balises autorisées dans une application XML et leur ordre, en spécifiant ainsi une « grammaire des balises » nommée DTD (définition du type du document). Une solution équivalente plus récente porte le nom de Schéma XML, et au contraire de la DTD, est écrite en XML ; toutefois, l'utilisation des DTD reste la plus répandue. Une DTD permet de définir et communiquer un jeu de balises à utiliser, en spécifiant notamment les noms de balises autorisés, leur enchaînement, les attributs autorisés, etc.

Ce sont les DTD qui, avec les spécifications écrites en langue naturelle, définissent les « applications d'XML ». Chaque application conçoit ses outils de traitement en fonction du sens des balises décrit dans sa spécification et de leur syntaxe précise décrite dans la DTD. Parmi les applications XML dans le domaine du traitement multilingue, citons plusieurs formats d'échange de ressources entre outils déjà existants. Ces formats permettent de structurer des données de telle sorte qu'elles soient lisibles indépendamment du logiciel utilisé. Ainsi, on a défini le format XLIFF[2] pour la « localisation », *i.e.* l'adaptation de documents ou programmes à une langue et à une culture données ; le format OLIF[3] pour les échanges de données lexicales et terminologiques ; le format TMX[4] pour l'échange, entre différents logiciels propriétaires, de mémoires de traduction (paires de syntagmes dans la langue source et la langue cible).

2 XLIFF : *XML Localization Interchange File Format*, www.xliff.org, format d'échange XML pour les fichiers à localiser.

3 OLIF : *Open Lexicon Interchange Format*, www.olif.net, format ouvert pour l'échange de lexiques.

4 TMX : *Translation Memory Exchange*, www.lisa.org/tmx, échange des mémoires de traduction.

### 2.1.3 La conversion des formats et l'affichage : XSLT

Les textes balisés en XML n'ont pas de format d'affichage intrinsèque. Pour les visualiser sous une forme quelconque, qui dépend des objectifs de l'application envisagée, il faut par exemple leur appliquer une feuille de style écrite en XSL[5], autre standard basé sur XML. Cette feuille de style, appliquée à travers un logiciel de conversion, changera les balises du fichier initial en un autre jeu de balises XML, ou en des balises HTML, ou même en un fichier texte. Des méthodes plus complexes existent également, permettant de produire des fichiers au format imprimable (PDF ou *PostScript*). On peut alors visualiser le résultat dans un navigateur classique – dans ce cas la feuille de style XSL joue le rôle d'une feuille de style CSS, utilisée pour HTML.

L'intérêt principal du mécanisme de feuilles de style XSL est que l'on peut définir de nombreux affichages différents (chaque feuille de style correspondant à un affichage) pour un même document en XML, selon la demande de celui ou de celle qui consulte la banque de données. On peut ainsi afficher tout ou partie d'un document (vue abrégée ou vue détaillée), afficher certaines explications, varier la langue dans laquelle on les affiche, et varier bien sûr l'aspect : couleurs, taille et définition de l'écran, etc.

### 2.1.4 La vérification et la validation

On dit d'un document balisé en XML qu'il est *bien formé* lorsque ses balises respectent les principes généraux de la syntaxe XML, tel l'emboîtement. On dit d'un document qu'il est *valide* si ses balises respectent *en plus* les règles définies dans une DTD. Le test de validité est donc plus contraignant que le test de bonne formation. On comprend en tout cas que pour qu'un document stocké dans une banque de textes puisse être utilisé par la suite – qu'il soit transformé pour être affiché selon divers formats, ou traité par différents outils logiciels – il est nécessaire qu'il soit valide.

De nombreux outils, plus ou moins simples, gratuits ou non, permettent de faire ces vérifications. Certains sont de simples vérificateurs autonomes, prenant peu de place en mémoire et offrant peu d'options, alors que d'autres font partie de boîtes à outils logicielles, dont ils constituent les éléments premiers. Les versions les plus récentes des navigateurs Internet courants effectuent également les tests de bonne formation : en ouvrant le document XML à tester dans un navigateur, le programme s'arrêtera sur la première erreur de bonne formation trouvée, s'il y en a une, ou affichera l'intégralité d'un document correct. Naturellement, lorsque le document est affiché, on voit en général l'ensemble des balises, sans formatage particulier, puisqu'il n'y en a pas par défaut (sans feuille de style). Plus récemment, les navigateurs peuvent aussi appliquer une feuille de style donnée à *n* fichier XML. Ces outils ne semblent pas, dans leur dernière version, vérifier la validité d'un document.

## 2.2 L'encodage des textes selon la spécification XCES

Le stockage des textes dans une banque de données nécessite en général l'utilisation d'un format spécifique, souvent fondé sur des *balises* insérées dans un texte, telles que définies initialement dans le langage SGML. La TEI (*Text Encoding Initiative*, initiative pour le codage des textes) a défini un tel jeu de balises pour le marquage des textes au format électronique, jeu spécifié initialement en SGML. Une simplification de la TEI est proposée dans le standard CES (*Corpus Encoding Standard*, standard de codage des corpus), qui s'inspire également des recommandations du projet européen EAGLES (*cf.* www.ilc.cnr.it/EAGLES96/home.html). CES a été récemment mis à jour et changé en XCES (*cf.* www.cs.vassar.edu/XCES/) pour être conforme au standard XML, en définissant grâce à une DTD des balises XML pour marquer les principales structures d'un texte.

Le modèle de données CES (donc XCES) définit deux principales classes d'annotation. D'abord, chaque document de la banque de textes contient un ensemble de méta-données, en format texte, structuré par des balises spécifiques à XCES. On peut indiquer ainsi un grand nombre d'informations utiles à propos du texte, dont un

5 XSL : *eXtended Stylesheet Language*, www.w3.org/Style/XSL/, langage de feuilles de style étendu ; plus spécifiquement, ici, XSLT, *i.e.* transformations XSL.

certain nombre sont obligatoires : par exemple, l'auteur, la date, un titre, la version, etc. Les balises fournies ici par la spécification XCES sont très nombreuses et permettent de structurer finement ces méta-données, pour inclusion plus tard dans un catalogue électronique, par exemple. Après cet en-tête, le corps du document contient le texte proprement dit, muni lui aussi de nombreuses annotations possibles. Celles-ci sont organisées en plusieurs niveaux : section ou chapitre, paragraphe, phrase ; à chaque niveau, un ensemble de phénomènes linguistiques et discursifs peuvent être annotés.

Nous proposons d'utiliser ici le standard XCES pour l'annotation des textes dans une banque. En effet, cette annotation est répandue dans le monde du traitement des corpus, et son respect de la syntaxe XML fait qu'elle peut être convertie facilement avec une feuille de style XSLT vers d'autres formats d'annotation, le cas échéant. De plus, le groupe de travail XCES met gratuitement à la disposition de la communauté une DTD (assez complexe), et des feuilles de style, pour transformer les ressources XCES. Pour commencer, on pourra utiliser seulement un sous-ensemble restreint de balises extraites du standard XCES[6].

## 3 La conversion de données textuelles vers le format XCES

Nous proposons dans ce qui suit un procédé de conversion vers XML et XCES des données textuelles qui peuvent être disponibles dans une institution ou une équipe.

### 3.1 Un procédé en trois étapes

Les principales étapes peuvent se résumer ainsi : (1) conversion des données existantes au format HTML, à l'aide du logiciel d'édition ayant servi à les créer ; (2) conversion des caractères dits « spéciaux » vers un jeu de caractères compatible avec XML (par exemple ISO-LATIN-1 ou UTF-8/Unicode) ; (3) conversion des balises dans les documents depuis HTML vers les conventions XCES ; (4) validation du document produit. Les documents ainsi obtenus peuvent être stockés dans une banque de textes ; en fonction du logiciel de gestion de la banque, une conversion des documents par des feuilles de style XSLT peut être nécessaire. Aussi, si l'affichage au format HTML est souhaité, les feuilles de style « XCES vers HTML » fournies avec XCES pourront être utilisées.

Parmi les étapes ci-dessus, la première est la moins coûteuse en temps et en ressources. La deuxième requiert un certain savoir-faire dans le domaine des jeux de caractères et nous présenterons ci-après (3.2) les principales conclusions d'un article paru antérieurement dans la présente revue (Chanard et Popescu-Belis 2001).

La troisième étape est la plus délicate, puisqu'il faut assigner les balises XCES aux différentes parties des documents, en fonction de leur « sens » – cette étape semble donc très difficile à automatiser. Les balises XCES étant plus riches (ou diverses) que les balises de formatage HTML, on ne peut savoir à l'avance comment encoder un texte donné dans le format XCES. On pourra commencer par appliquer un logiciel de « nettoyage » du code HTML (par exemple *HTML Tidy*[7]), et choisir d'abord des textes courts, ayant une structure assez simple. Parmi les opérations à accomplir, l'ajout de la déclaration XML est essentiel, et tient en quelques lignes seulement (prendre modèle sur des exemples fournis par XCES). Par ailleurs, il faut saisir un en-tête pour chaque document, dans le format XCES, ce qui suppose une bonne connaissance de la spécification XCES. Cet en-tête peut toutefois être simplifié pour des documents courts. On passera ensuite à un balisage simple et clair du corps du texte, que l'on pourra complexifier par la suite.

Une fois le document ainsi codé, la validation consiste à vérifier que le balisage suit celui spécifié dans la DTD de XCES. Pour ce faire, des outils gratuits de validation peuvent être téléchargés ; la section 3.3 décrit un ensemble d'éléments utiles pour la manipulation des fichiers XML et XCES.

6 Un manuel de balisage XCES, qui explique l'utilisation d'un sous-ensemble de balises XCES, a été utilisé dans la formation Rifal 2002-2003. Ce document a été élaboré par Marc Van Campenhoudt (ISTI, Bruxelles).

7 *HTML Tidy* : écrit par Dave Raggett, disponible gratuitement sur le site tidy.sourceforge.net.

### 3.2 L'encodage des caractères spéciaux : Unicode et XML

Si la conversion en HTML des ressources documentaires est souvent possible grâce aux logiciels d'édition, le problème des caractères dits « spéciaux » n'est pas toujours correctement résolu. Un caractère « spécial » peut être, par exemple, tout caractère qui ne figure pas dans l'ensemble de 26 lettres non accentuées de l'alphabet latin, ou plus souvent un caractère qui n'appartient pas à l'ensemble de caractères « courants » d'une région, tels le *o avec barre* (ø) ou le *s avec cédille* (ş) pour le monde francophone. On voit donc ce que cette notion comporte de relatif ; précisément pour cette raison, le standard Unicode met sur un pied d'égalité les différents alphabets grâce à un codage plus riche, contenant 65536 caractères.

Afin de réaliser des ressources interopérables, l'utilisation d'Unicode, prévue par défaut dans les documents XML, est vivement conseillée (Chanard et Popescu-Belis, 2001). Unicode permet en effet la représentation informatique aisée de multiples alphabets. Toutefois, le format d'affichage HTML généré par les logiciels ne suit pas forcément le codage Unicode, mais peut faire appel à des polices de caractères « locales » (balisage du type : *Mot en langue locale</font>*). Or, dans les documents XML, tous les caractères doivent se conformer au jeu de caractères déclaré au début du document XML, sans pouvoir passer par une information de police. Si l'on opte pour Unicode, on peut saisir directement les codes sur deux octets de chaque caractère (ou les générer grâce à un logiciel de conversion), ou bien insérer des entités avec le code hexadécimal, par exemple, en Unicode, *ə* pour le « *schwa* » (code hexadécimal 0259) ou bien *ə* pour le même caractère (code décimal 601).

### 3.3 Outils pédagogiques pour la conversion

Avant d'entreprendre la conversion d'un nombre important de ressources selon notre procédé, il est utile de le tester sur des documents relativement courts, et se familiariser avec les notions de bonne formation, validité, encodage, etc. Un outil simple de vérification, disponible gratuitement, est *RXP*[8] ; un outil gratuit pour l'encodage est *Recode*[9] ; un outil pour appliquer les feuilles de style est *Saxon*[10] ; rappelons également que les principaux navigateurs contiennent aussi des fonctionnalités pour XML. Dans la formation Rifal 2002-2003 que nous avons définie, un paquet pédagogique regroupait un certain nombre d'exercices utiles, ainsi que des logiciels et des exemples. À titre d'exercice, une application XML simple était proposée dans un premier temps : l'idée était de coder des fiches bibliographiques en XML, et de les transformer en HTML pour visualisation. Dans une second temps, des modèles de documents XCES (extraits du corpus Multext), la DTD de XCES et une feuille de style étaient introduits. Pour tous ces documents, on peut ainsi tester la bonne formation, la validité, et appliquer des feuilles de style. Notons toutefois ici que le jeu complet de feuilles de style XCES, conçu par les auteurs de XCES, nécessite un programme particulier de transformation XSLT, nommé *XT*[11].

## 4 Conclusion

L'alimentation des banques textuelles par la méthode décrite dans cet article se fonde sur plusieurs techniques élémentaires de manipulation de documents : production de documents XML munis de balises de type XCES, vérification de leur validité, application de feuilles de style XSLT données afin de générer des fichiers pour le transfert ou la visualisation.

L'ensemble de la méthode de conversion présentée ici peut paraître complexe et coûteuse. Il s'agit néanmoins de l'une des possibilités les plus directes pour la constitution de banques de textes multilingues à partir de documents existants rédigés dans des formats non contrôlés. Dans le cas de documents suffisamment importants, le coût de leur encodage correct en XCES est faible par rapport à la valeur de la ressource en elle-même, et les avantages apportés par

8 *RXP*, écrit par Richard Tobin, disponible à : www.cogsci.ed.ac.uk/~richard/rxp.html.

9 *Recode*, écrit par François Pinard, disponible à : www.iro.umontreal.ca/contrib/recode/HTML/.

10 *Saxon*, écrit par Michael Kay, disponible à : saxon.sourceforge.net.

11 *XT*, écrit par James Clark, disponible à : www.blnz.com/xt/.

l'existence d'une banque de textes, véritable bibliothèque minutieusement structurée, compense largement les efforts de conversion.

*Andrei Popescu-Belis*
*Institut pour les études sémantiques et cognitives (ISSCO), Unité de traitement informatique multilingue (TIM), École de traduction et d'interprétation, Université de Genève, Genève, Suisse.*
*andrei.popescu-belis@issco.unige.ch*

## Remerciements

Le contenu pédagogique de la formation Rifal 2002-2003 a été préparé par Andrei Popescu-Belis (ISSCO/TIM/ETI, Genève) pour l'Observatoire suisse des industries de la langue. L'auteur souhaite remercier pour leur aide Marc Van Campenhoudt (Isti, Bruxelles) Christian Chanard (LLACAN-CNRS, Villejuif), Marcel Diki-Kidiri (LLACAN, Villejuif), responsable du Groupe de travail formation du Rifal, ainsi que Marcel Grangier (Chancellerie fédérale suisse, Berne).

## Bibliographie

Amann (B.) et Rigaux (P.), 2002 : *Comprendre XSLT*, Paris : Éditions O'Reilly.

Chanard (C.) et Popescu-Belis (A.), 2001 : « Encodage informatique multilingue : application au contexte du Niger », *Cahiers du RIFAL*, nº 22, p. 33-45.

Consortium Unicode, 2000 : *The Unicode Standard Version 3.0*, Reading : Addison Wesley. Voir aussi www.unicode.org.

Harold (E. R.) et Scott Means (W. S.), 2001 : *XML in a Nutshell : a Desktop Quick Reference*, Sebastopol : O'Reilly & Associates.

Ide (N.), Bonhomme (P.) et Romary (L.), 2000 : « XCES : An XML-based Encoding Standard for Linguistic Corpora », Actes de LREC 2000 (2e Conférence internationale sur les ressources linguistiques et l'évaluation), Athènes, p. 825-830. Voir aussi www.cs.vassar.edu/XCES/.

ISO/IEC 10646-1, 2000 : *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1 : Architecture and Basic Multilingual Plane,* Genève : Organisation internationale de normalisation. Voir aussi www.iso.ch.

Popescu-Belis (A.), 2002 : « Apports d'Unicode à l'édition numérique multilingue », *Document numérique*, vol. 6, nº 3-4, p. 139-153.

Yang (J.), 2001 : « Comment construire une banque de terminologie véritablement multilingue », Actes du colloque *L'impact des nouvelles technologies sur la gestion terminologique*, Toronto.