Modélisation des connaissances dans le cadre de bibliothèques numériques spécialisées

Pierre-Edouard Portier*, Sylvie Calabretto *

*Université De Lyon, INSA de Lyon, LIRIS
pierre-edouard.portier,sylvie.calabretto@insa-lyon.fr

Résumé. Nous présentons une application innovante de la modélisation des connaissances au domaine des bibliothèques numériques spécialisées. Nous utilisons la spécification experte de la TEI (Text Encoding Initiative) pour modéliser la connaissance apportée par les chercheurs qui travaillent sur des archives manuscrites. Nous montrons les limites de la TEI dans le cas d'une approche diachronique du document, cette dernière impliquant la construction simultanée de structures de données concurrentes. Nous décrivons un modèle qui présente le problème et permet d'envisager des solutions. Enfin, nous justifions les structures arborescentes sur lesquelles se base ce modèle.

1 Introduction

Ce travail est une synthèse de l'expérience acquise à côtoyer des chercheurs qui construisent une édition électronique des archives manuscrites du philosophe Jean-Toussaint Desanti. L'édition électronique recouvre l'ensemble du processus non seulement éditorial mais scientifique et critique qui aboutira à la publication d'une ressource sous forme électronique. Pour les manuscrits, la première ressource publiée est un facsimile numérique auquel s'ajoutera le travail de transcription et d'analyse critique des chercheurs. Nous avons échangé avec les responsables d'autres projets similaires pour isoler la problématique centrale et partagée de la construction de documents multistructurés. Nous montrons dans la seconde section que cette problématique apparait naturellement avec l'utilisation de la TEI pour transcrire les manuscrits. Dans la troisième section nous proposons un modèle qui répond à cette problématique. Dans la dernière section nous formulons pour la première fois le problème de la construction de documents multistructurés.

2 La TEI et les documents multistructurés

2.1 Organisation de la TEI

La TEI, Text Encoding Initiative (Burnard et Bauman (2007)), est un consortium qui développe et maintient un standard pour la représentation des textes électroniques. Ses recommandations constituent une expertise dont peut profiter tout projet d'édition électronique. Elles sont exprimées sous la forme modulaire et extensible d'un schéma XML documenté.

2.2 Un sujet d'étude rebelle à la saisie hiérarchique

Un usage immédiat du formalisme XML impose une structure arborescente à son contenu, or les corpus des SHS ne se plient pas à cette contrainte structurelle. Un exemple récurrent est l'annotation de passages de discours direct qui chevauchent plusieurs paragraphes.

Au fil des recommandations de la TEI, pour chaque instance du problème, une solution locale est proposée. De plus, dans la dernière version des recommandations, un chapitre a été ajouté qui fait la synthèse de ces solutions ad-hoc. Elles sont au nombre de quatre (on trouvera une description plus détaillée dans DeRose (2004) et Bruno et al. (2007)).

Il est possible de dupliquer le contenu autant de fois qu'il y a de structures arborescentes concurrentes . . .solution pauvre qui empêche le document d'évoluer.

On peut utiliser des éléments vides ("fin de ligne" (<lb/>), "fin de page" (<pb/>), etc.). Il est alors impossible de profiter des outils standards de validation des documents XML, ou de construire un schéma (Simeon et Wadler (2003)) pour spécifier la structure qui utilise des éléments vides.

Pour deux structures concurrentes qui ne forment pas un arbre, on peut découper l'une d'entre elles en éléments suffisamment fins pour qu'ils ne chevauchent pas avec ceux de l'autre, et par le jeu d'attributs bien choisis, se laisser la possibilité de reconstruire la structure initiale. Comme les précédentes, cette solution demande des traitements particuliers pour reconstruire les différentes structures et ne permet pas l'utilisation des outils de typage XML.

Enfin, on peut isoler le contenu dans une structure de base et construire les structures documentaires par références à cette dernière. Cette solution est la plus générique, nous en proposons une variante dans la section suivante. D'autres variantes existent (Bird et Liberman (1999), Tummarello et al. (2005), Alink et al. (2006), Bruno et Murisasco (2007), Sperberg-McQueen et Huitfeldt (2000)) qui sont trop spécifiques pour supporter l'analyse que nous menons de la création des documents multistructurés.

3 $MultiX^2$, un modèle pour la représentation des documents multistructurés

3.1 MSDM, MultiX et $MultiX^2$

MSDM (Chatti et al., 2007) est un modèle pour la représentation de documents multi-structurés proposé par N.Chatti et dont une instance, MultiX, s'exprime dans le formalisme XML. Nous proposons une nouvelle instance, $MultiX^2$, qui privilégie les recommandations du W3C aux mécanismes spécifiques. Nous décrivons l'opération qui construit la structure de base à partir des structures documentaires.

3.2 $MultiX^2$

MSDM représente de façon optimale des structures documentaires. Soit une strophe du poème de Lewis Carroll "La chasse au snark", nous distinguons les structures des listings 1 et 2.

```
<lg><l>De ses hardes la perte importait fort peu, puisque</l>
    <l>Lors de son arrivee il portait sept vestons</l>
    <l>Et trois paires de chaussures; mais le pis est qu</l>
```

```
    <l>Il avait oublie totalement son nom</l>
</lg>
```

Listing 1 – *Structure poétique*

```
  <seg>De ses hardes la perte importait fort peu, puisque Lors de son arrivee il
       portait sept vestons Et trois paires de chaussures;</seg>
  <seg>mais le pis est qu Il avait oublie totalement son nom.</seg>
```

Listing 2 – *Structure logique*

Nous construisons la structure de base en identifiant les fragments partagés par les structures documentaires (voir listing 3).

```
<seg xml:id=''F1''>De ses hardes la perte importait fort peu, puisque</seg>
<seg xml:id=''F2''>Lors de son arrivee il portait sept vestons</seg>
<seg xml:id=''F3''>Et trois paires de chaussures;</seg>
<seg xml:id=''F4''> mais le pis est qu </seg>
<seg xml:id=''F5''>Il avait oublie totalement son nom</seg>
```

Listing 3 – *Structure de base*

Nous remplaçons le contenu des structures documentaires par des références à la structure de base grâce au standard XInclude (voir listings 4 et 5).

```
<lg>
  <l><xi:include href=''base.xml'' xpointer=''element(F1/1)''/></l>
  <l><xi:include href=''base.xml'' xpointer=''element(F2/1)''/></l>
  <l><xi:include href=''base.xml'' xpointer=''element(F3/1)''/>
     <xi:include href=''base.xml'' xpointer=''element(F4/1)''/></l>
  <l><xi:include href=''base.xml'' xpointer=''element(F5/1)''/></l>
</lg>
```

Listing 4 – *Structure poétique transformée*

```
  <seg><xi:include href=''base.xml'' xpointer=''element(F1/1)''/>
       <xi:include href=''base.xml'' xpointer=''element(F2/1)''/>
       <xi:include href=''base.xml'' xpointer=''element(F3/1)''/></seg>
  <seg><xi:include href=''base.xml'' xpointer=''element(F4/1)''/>
       <xi:include href=''base.xml'' xpointer=''element(F5/1)''/></seg>
```

Listing 5 – *Structure logique transformée*

Nous utilisons les outils XML standards pour valider les structures documentaires, et construire des requêtes grâce aux fonctions XQuery développées pour MultiX. Voir le listing 6 pour une requête qui trouve les vers coupés par une phrase.

```
let $poesie := doc(''poesie.xml'')
let $logique := doc(''logique.xml'')
for $p in $logique//seg,
    $v in $poesie//lg//l
where multix:share-fragments($p,$v) and not(multix:include-content-of($p,$v))
return $v
```

Listing 6 – *Exemple de requête*

La fonction share-fragments(a,b) vérifie si les éléments a et b ont au moins un fragment en commun, include-content(a,b) vérifie si tous les fragments qui composent l'élément b composent aussi l'élément a.

4 Une approche diachronique du document

De nombreux projets d'édition électronique partent des images de manuscrits puis les transcrivent et les annotent. Lors de ces opérations, il faudra gérer des cas de multistructure. Nous utilisons le modèle MultiX pour forger un point de vue propre à présenter le problème de la construction de documents multistructurés.

4.1 Les moments de restructuration

Nous analysons les conditions sous lesquelles il faut construire une nouvelle structure documentaire. Admettons que pour transcrire un manuscrit des chercheurs disposent des éléments de la TEI. Du temps passe avant qu'il ne soit impossible de baliser un fragment de texte sans créer de chevauchement. Si nous reprenons l'exemple précédent, des phrases et des vers sont convenablement balisés (voir figure 1) jusqu'à ce qu'une phrase chevauche un vers (voir figure 2). Il faut distinguer deux structures. La création d'une nouvelle structure est une opération purement formelle (voir figure 3) mais qui peut être l'occasion de choix d'un interprétant. Par exemple, demander à ce que la division en strophes rejoigne la structuration en vers (voir figure 4). Une étude de la trace de telles décisions permettrait de construire a posteriori des ontologies de domaine.

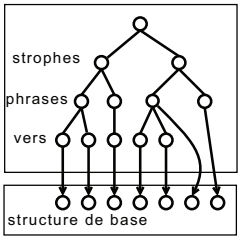

FIG. 1 – *Avant la restructuration*

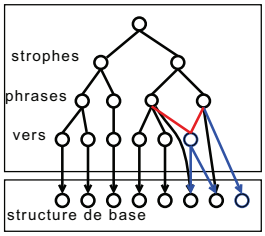

FIG. 2 – *Restructuration nécessaire*

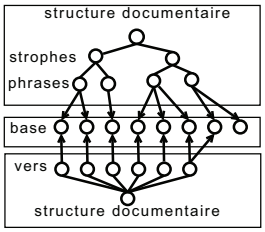

FIG. 3 – *Restructuration automatique*

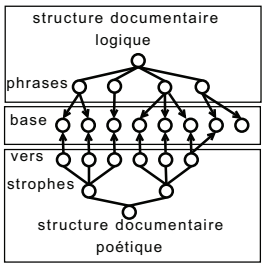

FIG. 4 – *Intervention de l'interprétant*

4.2 Le partage des responsabilités

Puisque les projets d'éditions électroniques de manuscrits s'inscrivent toujours dans un environnement multi-utilisateurs, il faut définir des rôles et des responsabilités. Nous supposons un responsable par structure.

Le responsable devra intervenir lors d'une modification de la structure de base. Nous distinguons cinq types de modifications (voir figure 5).

Dans la première situation, un fragment en lien avec la structure du responsable est modifié. Alerté, le responsable n'a pas de décision nécessaire à prendre.

Dans la seconde situation, un fragment de la structure de base en lien avec un élément de la structure du responsable est scindé. Le responsable doit être alerté car sa structure n'est plus cohérente. Il pourra choisir de regrouper les éléments scindés, de n'en conserver que quelques uns, etc.

Dans la troisième situation, des éléments d'autres structures coupent leurs liens avec un fragment de la structure de base relié à un des éléments de la structure du responsable. Le responsable doit être alerté à titre indicatif, en interprétant la situation, en interrogeant les autres responsables, etc. il pourra se construire une connaissance nouvelle.

Dans la quatrième situation, des fragments sans rapport avec la structure du responsable sont créés. Ce dernier doit être tenu informé pour éventuellement mettre à jour sa structure.

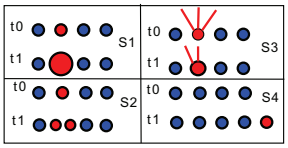

FIG. 5 – *Types de modifications de la structure de base*

4.3 Le choix des arbres

La détection des moments de restructuration utilise la contrainte qui définit un arbre comme une sorte de graphe, soit qu'un nœud fils ne peut avoir plus d'un père. Dans le contexte de l'édition électronique de manuscrits il ne semble pas que nous ayons d'autres propriétés algébriques utiles. L'usage de structures arborescentes est ainsi un moyen nécessaire pour automatiser le plus possible la construction de documents multistructurés.

5 Conclusions

Analysant les pratiques communes aux projets d'éditions électroniques de manuscrits, nous accédons au problème des documents multistructurés et proposons une solution grâce au modèle $MultiX^2$. Mais un nouveau problème apparait qui n'a jamais été énoncé : comment construire un tel document ? Nous avons éclairci cette problématique grâce à $MultiX^2$. Notons enfin que nous avons entamé une résolution concrète qui repose sur une correspondance entre les schémas XML et les types du langage de programmation fonctionnel Haskell (Jones, 2002), nous permettant d'assurer une bijection entre un document XML et les valeurs typées

qui existent à l'exécution d'un programme. Travailler dans l'espace des valeurs du langage facilite la gestion dynamique des structures.

Références

Alink, W., R. A. F. Bhoedjang, A. P. de Vries, et P. A. Boncz (2006). Efficient xquery support for stand-off annotation. In *XIME-P*.

Bird, S. et M. Liberman (1999). Annotation graphs as a framework for multidimensional linguistic data analysis. In M. Walker (Ed.), *Towards Standards and Tools for Discourse Tagging : Proceedings of the Workshop*, pp. 1–10. Somerset, New Jersey : Association for Computational Linguistics.

Bruno, E., S. Calabretto, et E. Murisasco (2007). Documents textuels multi structurés : un état de l'art. *Revue i3 7*(1).

Bruno, E. et E. Murisasco (2007). Msxd : A data model and an xquery extension for multi-structured xml documents. In *Session démonstrations, Actes des 23èmes Journées Bases de Données Avancées (BDA 2007)*. 6p.

Burnard, L. et S. Bauman (2007). Tei p5: Guidelines for electronic text encoding and interchange.

Chatti, N., S. Kaouk, S. Calabretto, et J.-M. Pinon (2007). MultiX: an XML-based formalism to encode multi-structured documents. In *Proceedings of Extreme Markup Languages'2007, Montréal (Canada)*.

DeRose, S. J. (2004). Markup overlap: A review and a horse. In *Extreme Markup Languages*.

Jones, S. P. (Ed.) (2002). *Haskell 98 Language and Libraries: The Revised Report*. http://haskell.org/.

Simeon, J. et P. Wadler (2003). The essence of xml. In *POPL '03: Proceedings of the 30th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, Volume 38, New York, NY, USA, pp. 1–13. ACM Press.

Sperberg-McQueen, C. M. et C. Huitfeldt (2000). Goddag: A data structure for overlapping hierarchies. In P. R. King et E. V. Munson (Eds.), *DDEP/PODDP*, Volume 2023 of *Lecture Notes in Computer Science*, pp. 139–160. Springer.

Tummarello, G., C. Morbidoni, et E. Pierazzo (2005). Toward textual encoding based on rdf. In M. Dobreva et J. Engelen (Eds.), *ELPUB*.

Summary

We describe an innovative application of knowledge modeling to the field of specialized digital libraries. We use the TEI (Text Encoding Initiative) specification to model knowledge of the researchers working on the archive. We show the inadequacy of the TEI specification when adopting a dynamic approach of the document. We present a model that allows us to specify this new problem.