Classement collaboratif de manuscrits

Pierre-Edouard Portier

Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205, F-69621, France
pierre-edouard.portier@insa-lyon.fr

RÉSUMÉ. *Pour chaque projet d'édition numérique de manuscrits, après que le corpus ait été constitué, les chercheurs commencent par le classer. Cette opération demande de grands efforts d'interprétation, elle n'est pas neutre mais contribue à la construction du point de vue du chercheur sur son objet d'étude. Ainsi, plusieurs classements peuvent être proposés pour un même sous-ensemble de l'archive. Or il n'existe pas de plateforme informatique spécifique pour assister les chercheurs dans cette opération délicate. Nous en proposons une sous la forme d'un service Web et d'une IHM qui prennent en compte les spécificités de la tâche de classement et peuvent profiter à tout projet qui étudie un corpus de documents numérisés dans un des domaines des Humanités.*

ABSTRACT. *Every electronic edition of manuscripts begins by ordering the studied corpus. This operation is of a highly hermeneutic nature and contribute to the construction of each researcher viewpoint. For a same fragment of the corpus, multiple orderings can be proposed. However, there is no digital plateform aimed at assisting researchers in this task. That is why we developed a Web service and a graphical interface dedicated to this operation and beneficial to any Humanities project.*

MOTS-CLÉS : *bibliothèques numériques, XML, classement, Haskell*

KEYWORDS: *digital libraries, XML, ordering, Haskell*

1. Introduction

De nombreux projets d'édition critique de manuscrits débutent par un classement du corpus des textes étudiés. Souvent les textes, de natures très diverses, sont initialement désordonnés. L'histoire des mains par lesquelles ils passèrent peut être complexe et un héritier les aura par exemple dérangés, ... Dans le cadre de notre projet de thèse financé par la région Rhône-Alpes, nous travaillons avec des chercheurs de l'ENS-LSH qui étudient les archives manuscrites du philosophe J.T.Desanti [1]. Fort de cette expérience, nous découvrons que l'opération de classement fait appel à la faculté d'interprétation. Le classement est ainsi toujours nécessaire *et* particulier. C'est pourquoi son instrumentation sera profitable. En effet, pour être propre à un interprétant elle demande d'être accompagnée de possibilités de travail collaboratif, pour être universelle elle demande à ce que son résultat soit partageable. La numérisation du corpus permet de travailler dans le domaine symbolique où il est facile de construire des classements concurrents de sous-ensembles de l'archive. L'existence d'un standard pour l'édition électronique de textes, la TEI ( ), permet d'assurer partage et réutilisation des résultats du classement. Notre travail consiste à déterminer et assurer les conditions pour permettre à plusieurs utilisateurs de travailler au classement d'un corpus d'archives manuscrites et pour offrir un accès unifié à une représentation standard des classements produits. Nous proposons une solution sous forme d'un service Web qui respecte le patron d'architecture dit REST ( ) afin d'inscrire les objets résultats du classement au sein même de la structure du Web et permettre de réutiliser ces objets dans une grande variété de contextes (publication, partage avec d'autres projets d'édition électronique, ...). Nous avons aussi développé un programme client écrit dans le dialecte Smalltalk Squeak ( ) qui offre une grande liberté d'interaction à l'utilisateur afin de simuler au mieux les opérations du chercheur à sa table (physique) de travail. Dans la suite, nous présenterons les travaux existants les plus proches de notre proposition, nous décrirons ensuite le service Web puis l'IHM.

2. État de l'art

2.1. *Propriétés nécessaires pour un système de classement d'archives manuscrites*

Après avoir interrogé et observé les chercheurs en sciences humaines avec lesquels nous travaillons, nous avons pu déterminer cinq caractéristiques nécessaires à tout système de classement :

– préservation de l'ordre initial de l'archive telle que trouvée avant numérisation

– système d'annotations évolué qui permette la création de relations n-aires (afin de pouvoir exprimer des assertions du type : "cette collection de pages est *une version alternative* de cette autre page")

1. environ 300 documents pour 30 000 pages

– environnement collaboratif où plusieurs utilisateurs peuvent proposer des classements concurrents.

– résultats du classement facilement publiables

Nous passons maintenant en revue les travaux existants qui répondent à un ou plusieurs des points précédents.

– Collate ( ) est un système Web de travail collaboratif orienté documents. Il ne permet pas le reclassement mais possède un système d'annotations collaboratives élaboré qui permet la construction d'une forme de discours. Cependant les annotations ne sont ici que des relations 1-aire. De plus, la publication des ressources n'est pas prise en compte.

– BAMBI ( ) et son successeur Diphilos ( ) sont des systèmes hypermedia pour la transcription de manuscrits. Les images de manuscrits sont entrées dans une base de données au début d'un projet et la construction de classements concurrents n'est pas possible. Cependant, la technologie SGML/HyTime utilisée permet la création de liens bidirectionnels entre pages, mais ne permet pas de gérer des relations n-aires. Finalement, il n'y a pas de possibilités simples de publier les ressources.

– Une partie du projet DEBORA ( ) consiste en une bibliothèque numérique avec des fonctionnalités de travail collaboratif. Y est introduit la notion de livre virtuel : la représentation d'un chemin à travers les pages de l'archive. Mais ces chemins ne sont pas eux-mêmes des ressources à part entière et ne peuvent pas entrer dans un processus collaboratif qui permettrait de les échanger, les annoter, etc.

– HyperNietzsche ( ) (aujourd'hui Nietzschesource) est un système pionnier de bibliothèque numérique. La problématique du reclassement est prise en considération très sérieusement mais le choix est fait de réaliser le classement une seule fois et par un petit comité d'experts. Comme pour DEBORA, un mécanisme de chemins existe qui a les mêmes défauts. TALIA ( ) est la suite d'HyperNietzche et utilise les technologies du Web sémantique, ce qui permet de représenter des relations n-aires.

– BRICKS ( ) est une architecture P2P de gestion de réseaux de bibliotèques numériques accompagné d'un ensemble d'applications construites au dessus de cette architecture. Elle introduit les deux notions de collections physiques et logiques. Les collections logiques contiennent des liens vers des objets de collections physiques. Mais un objet, en tant qu'il appartient à une collection logique, ne peut pas être annoté. RDF est utilisé pour créer des relations entre objets, il est donc possible de créer des relations n-aires.

Finalement, nous n'avons pas trouvé de solution qui réponde à l'ensemble des critères que nous avons énoncés plus haut. Ainsi, nous présentons maintenant une solution complète et générique à cette problématique.

3. Interface d'accès aux archives manuscrites

Dans cette partie nous décrivons un service Web qui permet de gérer des collections de pages manuscrites. Il assure une représentation toujours correcte de l'archive dans un sous-ensemble du langage XML défini par la TEI.

3.1. *Un standard : la TEI*

La TEI ( ), Text Encoding Initiative, 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.

Pour le classement d'archives manuscrites nous utilisons cinq balises de la TEI (graphic, teiCorpus, TEI, facsimile et surface ... voir figure 1 pour un exemple d'utilisation).

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <teiCorpus n="1">
3    <TEI>
4      <facsimile>
5        <surface n="1">
6          <graphic url="pochettes/0200/0001_high.jpeg"/>
7          <graphic url="pochettes/0200/0001_low.jpeg"/>
8        </surface>
9        <surface n="2">
10         <graphic url="pochettes/0200/0002_high.jpeg"/>
11         <graphic url="pochettes/0200/0002_low.jpeg"/>
12       </surface>
13       <surface n="3">
14         <graphic url="pochettes/0200/0003_high.jpeg"/>
15         <graphic url="pochettes/0200/0003_low.jpeg"/>
16       </surface>
17       <surface n="4">
18         <graphic url="pochettes/0200/0004_high.jpeg"/>
19         <graphic url="pochettes/0200/0004_low.jpeg"/>
20       </surface>
21     </facsimile>
22   </TEI>
23
24
25
26
```

```
28   <teiCorpus n="2">
29     <TEI>
30       <facsimile/>
31     </TEI>
32     <teiCorpus n="1">
33       <TEI>
34         <facsimile>
35           <surface n="1">
36             <graphic url="pochettes/0200/2_1/0001_high.jpeg"/>
37             <graphic url="pochettes/0200/2_1/0001_low.jpeg"/>
38           </surface>
39           <surface n="2">
40             <graphic url="pochettes/0200/2_1/0002_high.jpeg"/>
41             <graphic url="pochettes/0200/2_1/0002_low.jpeg"/>
42           </surface>
43           <surface n="3">
44             <graphic url="pochettes/0200/2_1/0003_high.jpeg"/>
45             <graphic url="pochettes/0200/2_1/0003_low.jpeg"/>
46           </surface>
47           <surface n="4">
48             <graphic url="pochettes/0200/2_1/0004_high.jpeg"/>
49             <graphic url="pochettes/0200/2_1/0004_low.jpeg"/>
50           </surface>
51         </facsimile>
52       </TEI>
53       <teiCorpus n="2">
```

Figure 1. *Partie d'un fichier XML TEI : les lignes 3 à 22 représentent un groupe de 4 pages qui entourent d'autres groupes, la description du premier de ces groupes commence à la ligne 28*

3.2. *Une architecture : REST*

La figure 2 est un modèle, sous forme de diagramme de classes UML, des ressources offertes par le service Web que nous avons développé. Notre service respecte le schéma d'architecture dit REST ( ). Le principe à la base de cette architecture est d'avoir un nombre illimité de ressources avec pour chacune un identifiant unique (par exemple une URL) et au plus les quatre opérations définies par le protocole HTTP : GET, PUT, DELETE et POST. De plus à chacune de ces opérations est associée une sémantique générique.

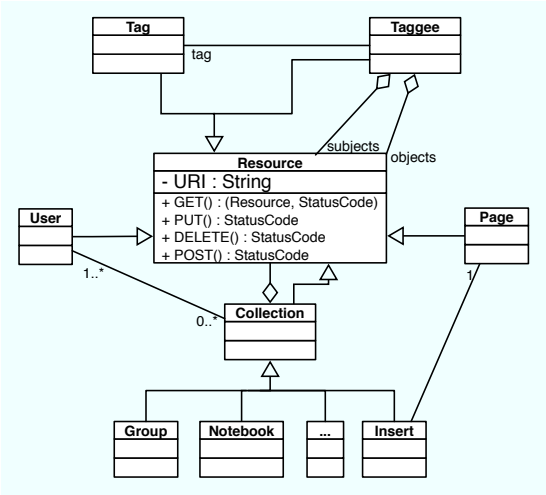

Figure 2. *modèle de domaine du service Web*

– La méthode GET doit servir à obtenir une représentation de la ressource, elle doit être sûre (ne pas modifier la ressource) et idempotente (elle produit toujours le même effet qu'elle soit itérée une ou plusieurs fois).

– La méthode PUT doit servir à modifier la représentation d'une ressource identifiée, si l'identifiant n'existe pas la ressource est créée. Cette opération est idempotente.

– La méthode DELETE doit servir à supprimer une ressource et être idempotente.

– La méthode POST doit être réservée aux opérations non sûres et éventuellement non idempotentes qui ne relèvent pas des trois premières méthodes.

Nos cinq types de ressources les plus essentiels sont l'utilisateur, le groupe, la page, la collection et l'insert. La collection est une agrégation de ressources, le groupe une collection d'utilisateurs, l'insert une sous-collection associée à la page de la collection où elle est insérée.

Le besoin de représentation de relations n-aires est satisfait grâce aux deux types de ressources *Tag* et *Taggee*. Un Tag est un terme. Un Taggee est une relation nommée d'un Tag et à laquelle des ressources peuvent participer soit en tant que sujets soit en tant qu'objets. Par exemple, l'assertion : "Les pages A et B sont des brouillons pour la page C" peut se modéliser au moyen d'un Taggee nommé du Tag "brouillon-pour" et avec trois participants : les pages A et B en tant que sujets, la page C en tant qu'objet.

Cette architecture en place, supposons que l'utilisateur bob veuille remplacer la troisième page de sa seconde collection par la première page du second des inserts qui se trouvent à la deuxième page du premier cahier de l'archive. Les opérations nécessaires sont GET suivi de PUT, informellement :

```
p := GET http://serveur.org/cahiers/1/pages/2/inserts/2/pages/1
PUT http://serveur.org/users/bob/collections/2/pages/3 p
```

Ou bien, supposons que l'utilisateur bob veuille ajouter la troisième page de la première collection de l'archive original à sa quatrième collection. Cette opération correspond à un GET suivi d'un POST, informellement :

```
p := GET http://serveur.org/collections/1/pages/3
POST http://serveur.org/users/bob/collections/4/pages/ p
```

3.3. *Traitement sûr des documents XML*

Comme rappelé dans ( ) il existe trois familles de solutions pour traiter des documents XML : les API XML telles que SAX ou DOM, les langages spécialisés tels que XSLT ou XDuce ( ), les isomorphismes entre types de données XML et types d'un langage de programmation. Les avantages de cette dernière solution associée à un langage de programmation fonctionnel pur (sans effets de bord) et fortement typé (nous choisissons Haskell) sont l'absence de phase d'analyse syntaxique et l'assurance offerte par le compilateur de tranformations correctes : un document valide se transforme en un document valide.

En combinant l'utilisation d'un sous-ensemble du langage de balisage défini par la TEI, une architecture REST et une correspondance entre types XML et types du langage Haskell nous avons développé un service Web de classement d'archives manuscrites qui est sûr, repose sur un standard établi et est "universellement" accessible car inscrit dans l'architecture même du Web.

4. IHM pour le classement d'archives manuscrites

4.1. *Utilisateurs cibles*

Nos utilisateurs sont des chercheurs en sciences humaines qui s'approprient un corpus documentaire en le classant. Ainsi, l'IHM doit offrir une grande liberté d'interactions pour simuler au mieux les opérations habituelles de ces chercheurs à leur table de travail. Les résultats de ces classements pourront devenir des objets consultables par d'autres types d'utilisateurs (simples lecteurs, philologues, etc.) au travers éventuellement d'une autre interface mais toujours servis par l'architecture décrite plus haut.

4.2. *Approche dynamique*

Nous nous sommes tournés vers le système de développement d'IHM appelé Morphic ( ) initialement développé pour le langage orienté objet par prototypes Self de Sun

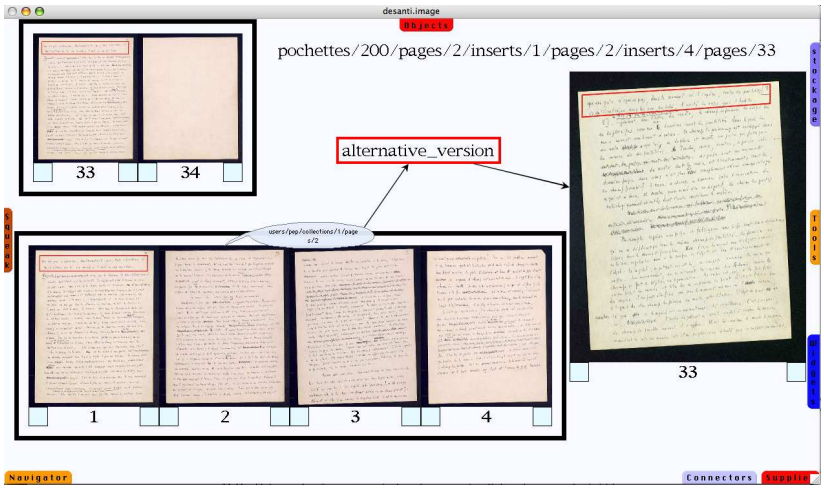

Figure 3. *Copie écran de l'IHM : en haut à gauche un navigateur ouvert sur l'ensemble de pages d'un insert d'une pochette ; en bas à gauche un navigateur ouvert sur une collection nouvellement créée pour classer les pages de l'insert précédent par glisser-déposer d'une collection dans l'autre ; sur la droite une page dont la collection nouvellement créée est une version alternative ; au centre la relation "version alternative" ; en haut un objet URL avec lequel beaucoup d'opérations sont possibles par simples glisser-déposer*

puis repris dans le dialecte Smalltalk Squeak ( ). Nous y avons trouvé les deux caractéristiques qui nous étaient nécessaires : interactivité et réactivité. La figure 3 est une copie écran de l'application. Le système est hautement interactif puisque pour accéder aux objets de l'interface, les examiner, changer leurs propriétés, il suffit d'une interaction directe avec leurs représentations graphiques sans qu'il soit nécessaire de passer par une représentation intermédiaire. Nous obtenons ainsi la vitesse de développement dont nous avions besoin.

5. Conclusions

Les chercheurs en sciences humaines et sociales travaillent principalement sur des corpus de textes. Ces derniers nécessitent presque toujours un classement. Cette opération est très délicate et requiert toute l'attention et l'imagination des chercheurs. Il n'existait pas d'outil numérique spécifique pour assister les chercheurs dans ce classement pourtant systématiquement nécessaire. Nous avons développé cet outil et l'avons conçu suffisamment générique et interopérable pour répondre aux besoins des chercheurs des diverses disciplines (philosophie, littérature, etc.) qui traitent des corpus textuels et pour leur permettre de partager facilement les résultats de leur travail. Finalement, nous sommes en train d'inscrire ce programme au sein d'une plateforme que

nous développons et qui est destinée à assister les chercheurs du domaine des Humanités dans l'édition numérique de textes ; elle comprend, entre autres fonctionnalités, un module évolué pour la transcription.

6. Bibliographie

[ATA 03] ATANASSOW F., CLARKE D., JEURING J., « Scripting XML with Generic Haskell », rapport, 2003, Utrecht University.

[BER 07] BERTONCINI M., « On the Move Towards the European Digital Library : BRICKS, TEL, MICHAEL and DELOS Converging Experiences », *Research and Advanced Technology for Digital Libraries, 11th European Conference, ECDL 2007, Budapest, Hungary, September 16-21, 2007, Proceedings*, vol. 4675 de *Lecture Notes in Computer Science*, Springer, 2007, p. 440-441.

[BOZ 97] BOZZI A., CALABRETTO S., « The Digital Library and Computational Philology : The BAMBI Project », *ECDL*, vol. 1324 de *Lecture Notes in Computer Science*, Springer, 1997, p. 269-285.

[BOZ 04] BOZZI A., « The DIPHILOS workstation for critical apparatus management : some experiments on medieval provencal texts », *Textual Criticism and Genetics Confronting Methods, Louvain-la-Neuve*, 2004.

[BUR 07] BURNARD L., BAUMAN S., « TEI P5 : Guidelines for Electronic Text Encoding and Interchange », , 2007.

[D'I 07] D'IORIO P., « Nietzsche on New Paths : The HyperNietzsche Project and Open Scholarship on the Web », *Maria Cristina Fornari, Sergio Franzese (Ã©ds.), Friedrich Nietzsche. Edizioni e interpretazioni, Pisa ETS*, 2007.

[FIE 00] FIELDING R. T., « Architectural styles and the design of network-based software architectures », PhD thesis, 2000.

[HAH 08] HAHN D., NUCCI M., BARBERA M., « The Talia library platform - Rapidly building a digital library on Rails », *4th Workshop on Scripting for the Semantic Web*, 2008.

[HOS 00] HOSOYA H., PIERCE B. C., « XDuce : A typed XML processing language (preliminary report », *In Proc. of Workshop on the Web and Data Bases (WebDB*, Springer-Verlag, 2000, p. 226–244.

[ING 97] INGALLS D., KAEHLER T., MALONEY J. M., WALLACE S., KAY A., IMAGINEERING W. D., « Back to the future : The story of Squeak, A practical Smalltalk written in itself », *In Proceedings OOPSLA 97, ACM SIGPLAN Notices*, ACM Press, 1997, p. 318–326.

[MAL 95] MALONEY J. H., SMITH R. B., « Directness and Liveness in the Morphic User Interface Construction Environment », *In Proceedings of User Interface and Software Technology (UIST 95) ACM*, ACM Press, 1995, p. 21–28.

[NIC 00] NICHOLS D. M., PEMBERTON D., DALHOUMI S., LAROUK O., BELISLE C., TWIDALE M. B., « DEBORA : developing an interface to support collaboration in a digital library », *European Conference on Digital Libraries*, Springer, 2000, p. 239–248.

[STE 04] STEIN A., KEIPER J., BEZERRA L., BROCKS H., THIEL U., « Collaborative Research and Documentation of European Film History : The COLLATE Collaboratory », *In International Journal of Digital Information Management*, 2004, p. 30–39.