Cahiers GUTenberg

EXPÉRIENCE DE CODAGE DE DOCUMENT À INTÉRÊT GRAPHIQUE À L'AIDE DE TEI

Jean-Daniel FEKETE

Cahiers GUTenberg, nº 28-29 (1998), p. 131-142.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1998___28-29_131_0>

Expérience de codage de document à intérêt graphique à l'aide de TEI

Jean-Daniel Fekete*

École des Mines de Nantes
4, rue Alfred Kastler, La Chantrerie
BP 20722, 44307 NANTES Cedex 3, France
Jean-Daniel.Fekete@emn.fr

Résumé. Alors que le codage numérique de documents purement textuels est maintenant bien maîtrisé, les documents textuels à intérêt graphique posent encore de nombreux problèmes.

Dans cet article, nous décrivons la façon dont nous avons codé l'encyclopédie *La chose imprimée* en SGML à l'aide d'une adaptation de la DTD TEI.

Dans cette encyclopédie, certains articles décrivent des règles de mise en page ou de typographie en les appliquant au texte de la description lui-même. En ne codant que le texte du document — c'est-à-dire en perdant la présentation originelle des exemples de mise en page ou de typographie — il perd tout son intérêt. En ne gardant que l'image du document, il perd les vertus du texte (indexation, recherche rapide, présentations multi-vues, etc). La difficulté a donc été de coder le document textuel en relation avec ses traits graphiques et typographiques pertinents.

Une fois ce document codé, nous avons réalisé des traducteurs de TEI vers LaTeX et HTML. Le respect de la présentation originelle pose de problèmes de traduction spécifiques qui ne sont pas pris en compte par les traducteurs existant.

En effet, la voie normale pour le traitement des documents SGML est DSSSL, qui ne dispose pas encore des mécanismes dont nous avons besoin pour gérer les documents textuels à intérêts graphiques. Nous avons donc dû développer les programmes de traduction en PERL.

1. Introduction

Les bibliothèques électroniques se développent rapidement, permettant l'accès distant à un nombre croissant de documents. La numérisation de documents

* Ce travail a été partiellement financé dans le cadre du projet Emma du GSI Cognition.

textuels ne pose pas de problèmes techniques : il existe plusieurs conventions de codage permettant de les stocker et de les rendre accessibles sans perte d'information [10]. En revanche, pour les documents textuels à intérêts graphiques, le problème du codage n'est pas résolu.

Nous considérons qu'un document textuel a un intérêt graphique si son intérêt est autant lié à son contenu textuel qu'à la présentation originelle du texte. C'est le cas des manuscrits, des livres anciens, des manuels de typographie pour ne citer que ceux-là. Dans cet article, nous nous intéressons au codage de ces documents. Dans la section qui suit, nos décrivons précisément ce que nous voulons faire en termes de propriétés sur le document électronique. Ensuite, nous faisons un rapide état de l'art sur les méthodes de description de la présentation des documents. Nous décrivons ensuite notre approche et montrons quelques résultats. La dernière section présente quelques perspectives.

2. Spécification déclarative de l'apparence

Il existe une grande variété de documents textuels à intérêts graphiques. On peut classer sur deux dimensions la complexité du codage de leur apparence : la précision et la difficulté.

Par exemple, le projet des textes de femmes écrivains de l'université de Brown aux USA (WWP) [7] impose que certains aspects de la présentation originelle soient codés. Cependant, ce codage n'est pas destiné à produire une apparence fidèle du document mais plutôt à évoquer cette apparence originelle.

À l'autre extrême, le codage des manuscrits pour leur étude génétique [6] est à la fois complexe et nécessite une grande précision.

Entre ces deux extrêmes existe un continuum de documents, dont les livres imprimés traitant de typographie qui nous intéressent ici. En particulier, l'Encyclopédie de la chose imprimée [4] et le manuel typographique de Fournier [5] qui présentent un nombre important de difficultés de codage.

2.1. Propriétés désirées du codage de l'apparence

Pour qu'un codage de l'apparence originelle soit utile, il faut qu'il :

1. permette de reformater le document codé de façon similaire à sa forme initiale,
2. fasse partie intégrante du document codé,
3. autorise le raisonnement sur l'apparence du document, et
4. facilite la factorisation des descriptions récurrentes.

Le point (1) permet de vérifier la qualité du codage. Le point (2) signifie que la présentation originelle *fait partie* du document. Le point (3) facilite les recherches de propriétés sur le document. Par exemple, dans l'encyclopédie de la chose imprimée, la maquette n'est pas cohérente tout au long du livre à cause de la réutilisation de parties de la première édition dans la deuxième édition. Un bon codage doit permettre de retrouver les pages provenant de la première édition. Le point (4) permet de simplifier la tâche de codage et d'analyse. Il existe déjà des langages permettant de transformer une structure logique en structure d'affichage. Nous allons passer en revue les principaux avant de décrire notre choix.

3. Gestion de l'apparence

Il existe plusieurs solutions pour coder l'apparence d'un texte. Nous avons un grand choix entre les langages de description de pages comme PostScript ou PDF, les systèmes de formatage à la LaTeX ou troff, ou les formats de documents structurés comme SGML. Il est maintenant admis dans la communauté des bibliothèques électroniques que SGML offre les meilleurs compromis [9].

Les techniques utilisées aujourd'hui pour donner une apparence à un document structuré sont basées principalement sur les feuilles de styles [8]. Seule la TEI dispose d'un mécanisme censé décrire la présentation originelle, en utilisant l'attribut spécifique.

3.1. Les feuilles de styles

Les feuilles de styles permettent de traduire une structure logique de document en une structure d'affichage, destinée à être imprimée ou affichée sur un écran. Il existe schématiquement deux types de langages de feuilles de styles : déclaratives et procédurales. Les premières tentent de décrire la transformation entre la structure d'un document et sa présentation par des règles de transformation déclaratives, tandis que les deuxièmes utilisent des procédures pour indiquer le traitement à effectuer sur les éléments structurels afin de les afficher.

Traditionnellement, l'avantage des premiers est la relative simplicité des règles. En revanche, les feuilles de styles procédurales permettent des descriptions beaucoup plus sophistiquées de présentation. Plus formellement, les feuilles de styles déclaratives ne permettent pas d'avoir une structure physique de présentation très éloignée de la structure logique du document, au contraire des feuilles de styles procédurales. En revanche, les feuilles de style procédurales ne peuvent être faites que par des programmeurs.

Une description exhaustive des langages de feuilles de styles dépasse le cadre de notre article. On peut se rapporter à [8] pour une comparaison plus approfondie. Parmi les langages disponibles, lesquels vérifient les propriétés que nous avons définies dans la section précédente ? Nous avons considéré CSS [3], XSL [1] et DSSSL [2].

À titre de comparaison, voici le code nécessaire pour spécifier que le contenu de l'élément logique `<EMPH>` doit se composer en gras :

CSS	XSL	DSSSL

`EMPH {`
` font-weight: bold;`
`}`
	`<rule>`
	` <target-element`
	`  type="EMPH"/>`
	`  <SPAN`
	`  font-weight="bold">`
	`  <children/>`
	`  </SPAN>`
	`</rule>`
		`(element EMPH`
		` (make sequence`
		`  font-weight: 'bold`
		`  (process-children-trim)))`

La propriété (1) est vérifiée de tous les langages décrivant une présentation, sauf CSS, qui ne permet pas de décrire toutes les caractéristiques des livres imprimés (notes en bas de page ou marginales par exemple). La propriété (2) n'est vérifiée par aucun langage de style, sauf partiellement par CSS. Il doit cependant être possible d'inclure des styles XSL à l'intérieur d'un document, dans son en-tête par exemple. DSSSL utilise une syntaxe trop particulière qui rend difficile, voire impossible, le point (3). CSS rend difficile le point (4). XSL serait donc utilisable, mais n'était pas spécifié au début de notre travail. De plus, XSL repose sur XML et TEI repose encore sur SGML, ce qui les rend incompatibles pour le moment. Nous avons donc utilisé un autre système de spécification de présentation tirant profit du format TEI.

3.2. Le format TEI

La TEI *(Text Encoding Initiative)* est un projet visant à faciliter l'échange de documents entre chercheurs en sciences sociales. Un comité composé de scientifiques de tout premier rang a défini un ensemble de règles de codage facilitant les échanges. Les détails dépassent largement l'objet de cet article[1]. Cette section décrit simplement la façon dont la TEI structure un document.

Chaque document conforme à la TEI est organisé de la façon suivante :

1. Voir le *Cahier GUTenberg* 24, juin 1996, consacré à la TEI.

```
<TEI.2>
    <teiHeader> [ informations contenues dans l'en-tête TEI ]
    </teiHeader>,
    <text>
        <front>[ textes préliminaires... ] </front>,
        <body>[ corps du texte... ] </body>
        <back> [annexes... ] </back>
     </text>
</TEI.2>
```

Le codage du document lui-même est placé après la balise `<text>`. Notons ici que TEI ne définit pas *une* DTD mais une famille de DTDs. Chaque projet de codage peut décider de puiser parmi un ensemble de composantes formant une DTD spécifique adaptée au codage du document.

Quelle que soit la DTD utilisée, la TEI définit un attribut standard *REND* destiné à indiquer la façon dont le document originel était composé (caractères gras, italique, etc). Par exemple, une indication scénique dans une pièce de théâtre pourrait se coder ainsi :

```
<stage rend=italic>
Enter Barnardo and Francisco, two Sentinels, at several doors</stage>
```

L'attribut *REND* spécifie que cette indication scénique était composée en italique dans le document originel.

L'en-tête est peut-être la partie la plus originale de la TEI, car elle force toute personne désirant coder un document à décrire de manière très précise les règles utilisées. Cet en-tête est fait pour favoriser l'échange des documents. Une partie de l'en-tête est prévue pour décrire les conventions utilisées pour coder la composition originelle du document, appelée *rendition*. Il est donc en principe possible de décrire dans l'en-tête la façon dont l'apparence est codée. Cependant, les recommandations de la TEI ne définissent pas la sémantique des éléments *rendition* ni du contenu de l'attribut *REND* ! Nous allons en proposer une interprétation.

4. Approche utilisée

Nous avons donc utilisé la convention mise au point par le WWP [7] pour coder la présentation originelle. Cette convention donne une sémantique à l'attribut *REND* de TEI, similaire à l'attribut *STYLE* utilisé par CSS. L'attribut *REND*

contient une chaîne de caractères avec une alternance mot-clef/valeur. La valeur est toujours entourée de parenthèses, ce qui facilite l'analyse lexicale et permet d'avoir des valeurs par défaut. En effet, un mot-clef suivi d'un autre mot-clef associe la valeur (1) au premier mot-clef.

Chaque mot-clef est un attribut de présentation associé à l'élément TEI. Les attributs reconnus sont décrits dans la section suivante. Nous indiquons ensuite la façon dont le document peut décrire sa présentation de manière relativement compacte.

4.1. Utilisation de l'attribut REND

Les attributs définis par le WWP sont les suivants :

ALIGN (B) `left, right, center`
BEQUEATH (BL) idem, mais limite la portée de la modification.
BESTOW (BL) modifie le rendu implicite d'une liste d'éléments
BREAK (L) indique si l'élément commence sur une nouvelle ligne
CASE (BL) `allcaps, lower, smallcaps, upper, mixed`
COLUMNS (B) nombre de colonnes
FACE (BL) nom d'une famille de polices
FILL (L) texte utilisé répétitivement pour remplir l'espace de l'élément précédent à l'élément courant
FIRST-INDENT (B) position de la marge gauche de la première ligne d'un paragraphe
FONT (BL) identificateur d'une police
GET (BL) utilisation d'un attribut de rendu indirect
INDENT (B) position de la marge de gauche par rapport au bord gauche de la page
KERN (BL) espacement entre les lettres
LINELEN (BL) largeur de ligne
POST (BL) texte à insérer littéralement après l'élément
PRE (BL) texte à insérer littéralement avant l'élément
RIGHT-INDENT (B) position de la marge de droite par rapport au bord droit de la page
SIZENAME (BL) nom d'une taille de caractères (pica, etc.)
SIZE (BL) taille des caractères, en relatif (avec $+$, $++$, $-$ ou $--$) ou en absolu
SLANT (BL) `upright, italic, oblique, reverse-oblique`
SPACE-AFTER (B) espacement vertical après l'élément
SPACE-BEFORE (B) espacement vertical avant l'élément
SUB (L) indice
SUP (L) exposant
WEIGHT (BL) `light, normal, bold`

Nous avons ajouté les attributs suivants :

DISPLAY (BL) `block, inline` spécifie si l'élément doit être placé dans un bloc à part ou en ligne.
EVEN-MARGIN (P) largeur de la marge gauche des pages paires
FOOT-SEP (P) distance du bas du texte principal au haut de l'en-pied.
HEAD-HEIGHT (P) hauteur de l'en-tête des pages
HEAD-SEP (P) distance de l'en-tête au haut du texte principal
HRULE-AFTER (B) épaisseur du filet placé après l'élément
HRULE-BEFORE (B) épaisseur du filet placé avant l'élément
LINE-SPACING (B) distance entre les lignes
MARGINAL-SEP (P) distance du texte principal à la marge des notes
MARGINAL-WIDTH (P) largeur des notes en marge
ODD-MARGIN (P) largeur de la marge gauche des pages impaires
PAGE-HEIGHT (P) hauteur de la page
PAGE-WIDTH (P) largeur de la page
TEXT-HEIGHT (P) hauteur du texte principal
TEXT-WIDTH (P) largeur du texte principal
TOP-MARGIN (P) largeur de la marge du haut

Bien entendu, tous les attributs ne sont pas utilisables avec tous les éléments. Il existe une distinction initiale entre les éléments *en-ligne* (marqués d'un L ci-dessus) et les éléments *en-bloc* (marqués d'un B). Les premiers font partie du texte tandis que les seconds définissent des blocs de texte. De plus, les attributs de page (P) ne peuvent être définis que pour l'élément de passage à la page suivante `<PB>`.

4.2. Spécification déclarative de la présentation

Les éléments `<rendition>` et `<tagusage>` permettent de spécifier la présentation par défaut des éléments composant un document. Dans la version actuelle, la spécification se fait élément par élément. Pour pallier cette limitation, il existe un mécanisme pour modifier dynamiquement la présentation d'une liste d'éléments dans un contexte particulier. Par exemple, pour exprimer que les items d'une liste commencent par un point, l'en-tête TEI contiendrait le codage suivant :

```
<rendition rend='pre (. ) bestow ((pre (- )) (item))'
 id=item_defaut>présentation d'un item par
défaut</rendition>
<tagusage gi=item render=item_defaut></tagusage>
```

L'attribut *PRE* indique le préfixe à appliquer à chaque item. L'attribut *BESTOW* permet de changer dynamiquement la valeur d'un attribut. Dans cet exemple, il indique que l'attribut de présentation *PRE* prendra la valeur `(- )` à

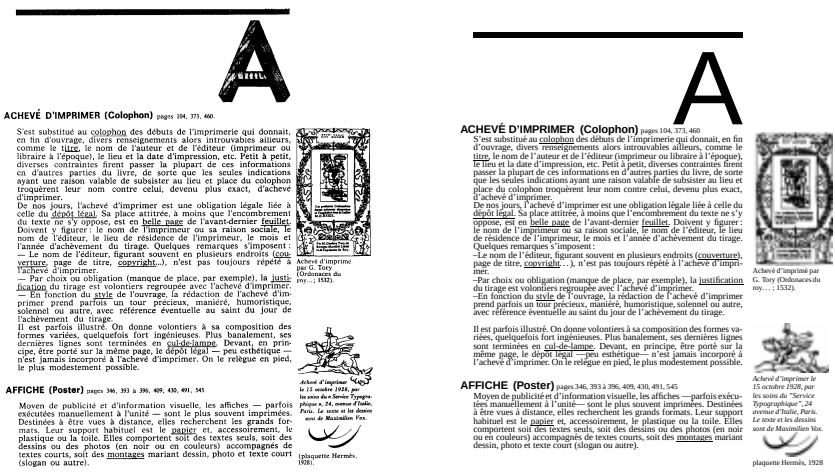

FIGURE 1 – *Comparaison de la première page de l'encyclopédie de la chose imprimée, dans sa version originelle à gauche et dans la version reformatée à droite*

l'intérieur d'un élément `<item>`. Ce mécanisme permet une spécification contextuelle de la présentation sans mécanisme de *pattern matching* généralement utilisé dans les feuilles de styles. Par exemple, CSS permet d'exprimer qu'une présentation ne sera active que dans un contexte donné (ici, lorsqu'une liste est à l'intérieur d'une autre liste) :

```
UL { list-style: dash outside }
UL ~ UL { list-style: circle outside }
```

5. Résultats

La Figure 1 compare la version originelle et reconstruite de la première page de l'encyclopédie. La différence est visible, mais la similitude est évidente. Dans les détails, les différences apparaissent, causées par des caractères différents et des algorithmes de composition différents. Ce résultat a été obtenu avec le programme `tei2latex`, qui traite un document TEI et génère du LaTeX, comme son nom l'indique. Nous disposons à la fois du document codé et de l'image

```
<div0 id=a n=A type=division><head>A</head>

<entry id=defacheve type=definition>
<form>ACHEVÉ D'IMPRIMER</form>
<trans><tr lang=en>Colophon</tr></trans>
<xr><ptr target='col104 ach373 ach460'></xr>
<note place=margin>
<xfigure doc=lci15 from='space (2d) (124 95) (153 144)'>
<head>Achevé d'imprimé par <name>G. Tory</name>
(<mentioned lang=oldfr>Ordonaces du
roy</mentioned>...; 1532).</head>
</xfigure>
</note>
<encycl>
<p>S'est substitué au <ref type=def target=defcolophon
id=colophon15>colophon</ref>des débuts de l'imprimerie
qui donnait, en fin d'ouvrage, divers renseignements alors
introuvables ailleurs, comme le <ref type=def target=deftit
id=tit15>titre</ref>,
le nom de l'auteur et de l'éditeur (imprimeur ou libraire à
l'époque), le  lieu et la date d'impression, etc. Petit à
petit, diverses contraintes firent passer la plupart de ces
informations en d'autres parties du livre, de sorte que les
seules indications ayant une raison valable de subsister
au lieu et place du colophon troquèrent leur nom contre
celui, devenu plus exact, d'achevé d'imprimer.</p>
<p>De nos jours, l'achevé d'imprimer est une obligation
légale liée à celle du <ref type=def target=defdep
id=dep15>dépôt légal</ref>. Sa place attitrée, à  moins que
l'encombrement du texte ne s'y oppose, est en <ref type=def
target=defbellepage id=bellepage15>belle page</ref> de
l'avant-dernier <ref type=def  target=deffeu
id=feu15>feuillet</ref>. Doivent y figurer : le nom de
l'imprimeur ou sa raison sociale, le nom de l'éditeur, le
lieu de résidence de  l'imprimeur, le mois et l'année
d'achèvement du tirage. Quelques remarques s'imposent :</p>

<list>
<item>Le nom de l'éditeur, figurant souvent en plusieurs
endroits (<ref type=def target=defcouverture
id=couverture15>couverture</ref>),
page de titre, <ref type=def target=defcopyright
id=copyright15>copyright</ref>...), n'est pas toujours
répété à l'achevé d'imprimer.</item>
<item>Par choix ou obligation (manque de place, par
exemple), la <ref type=def target=defjus
id=jus15>justification</ref> du tirage est volontiers
regroupée avec l'achevé d'imprimer.</item>
<item>En fonction du <ref type=def target=defsty
id=sty15>style</ref> de l'ouvrage, la rédaction de l'achevé
d'imprimer prend parfois un tour précieux, maniéré,
humoristique, solennel ou autre, avec référence éventuelle
au saint du jour de l'achèvement du tirage.</item>
</list>
<note place=margin>
<xfigure doc=lci15 from='space (2d) (125 51) (153 69)'>
<head rend=italic>Achevé d'imprimer le <date>15 octobre
1928</date>, par les soins du <Q>Service Typographique</Q>,
24 avenue d'Italie, Paris. Le texte et les dessins sont de
<name>Maximilien Vox</name>.</head>
</xfigure>
</note>
<p>Il est parfois illustré. On donne volontiers à sa
composition des formes variées, quelquefois fort
ingénieuses. Plus banalement, ses dernières lignes sont
terminées en <ref type=def target=defculdelampe
id=culdelampe15>cul-de-lampe</ref>. Devant, en principe,
être porté sur la même page, le dépôt légal &mdash;peu
esthétique&mdash; n'est jamais incorporé à l'achevé
d'imprimer. On le relègue en pied, le plus modestement
possible.</p>
</entry><!--ACHEVÉ D'IMPRIMER-->
```

FIGURE 2 – *Codage TEI de la première page de «La chose imprimée»*

de chaque page, ce qui nous permet d'extraire les illustrations : le document codé ne contient que des références externes vers les zones d'illustrations. La Figure 2 montre le codage TEI de la page.

La traduction en HTML est beaucoup moins intéressante, car HTML ne peut pas intrinsèquement reproduire des notes en bas de page ou marginales. Nous avons néanmoins réalisé une version de notre traducteur essayant de gérer «au mieux» les spécifications de présentation. À titre d'information, la Figure 5 montre la même page.

5.1. Problèmes

La version actuelle de notre traducteur a mis en évidence plusieurs points intéressants :

- les outils actuels de transformation de SGML/XML vers une présentation ne sont pas assez sophistiqués pour accepter une spécification de présentation *dans* le document ;
- la spécification de présentation de la WWP, même améliorée, n'est pas assez expressive pour l'encyclopédie de la chose imprimée.

IMAGE TOP UP PREVIOUS NEXT

1 ACHEVÉ D'IMPRIMER

(Colophon)

Achevé d'imprimé par G. Tory (*Ordonaces du roy...*; 1532).

S'est substitué au colophon des débuts de l'imprimerie qui donnait, en fin d'ouvrage, divers renseignements alors introuvables ailleurs, comme le titre, le nom de l'auteur et de l'éditeur (imprimeur ou libraire à l'époque), le lieu et la date d'impression, etc. Petit à petit, diverses contraintes firent passer la plupart de ces informations en d'autres parties du livre, de sorte que les seules indications ayant une raison valable de subsister au lieu et place du colophon troquèrent leur nom contre celui, devenu plus exact, d'achevé d'imprimer.

De nos jours, l'achevé d'imprimer est une obligation légale liée à celle du dépôt légal. Sa place attitrée, à moins que l'encombrement du texte ne s'y oppose, est en belle page de l'avant-dernier feuillet. Doivent y figurer : le nom de l'imprimeur ou sa raison sociale, le nom de l'éditeur, le lieu de résidence de l'imprimeur, le mois et l'année d'achèvement du tirage. Quelques remarques s'imposent :

- Le nom de l'éditeur, figurant souvent en plusieurs endroits (couverture), page de titre, copyright...), n'est pas toujours répété à l'achevé d'imprimer.
- Par choix ou obligation (manque de place, par exemple), la justification du tirage est volontiers regroupée avec l'achevé d'imprimer.
- En fonction du style de l'ouvrage, la rédaction de l'achevé d'imprimer prend parfois un tour précieux, maniéré, humoristique, solennel ou autre, avec référence éventuelle au saint du jour de l'achèvement du tirage.

Il est parfois illustré. On donne volontiers à sa composition des formes variées, quelquefois fort ingénieuses. Plus banalement, ses dernières lignes sont terminées en cul-de-lampe. Devant, en principe, être porté sur la même page, le dépôt légal -peu esthétique- n'est jamais incorporé à l'achevé d'imprimer. On le relègue en pied, le plus modestement possible.

Achevé d'imprimer le 15 octobre 1928, par les soins du "Service Typographique", 24 avenue d'Italie, Paris. Le texte et les dessins sont de Maximilien Vox.

IMAGE TOP UP PREVIOUS NEXT

FIGURE 3 – *Même page qu'en Figure 1 mais traduite en HTML*

Nous avons dû utiliser le langage PERL [11] pour réaliser un traducteur de TEI vers LaTeX. Il a ainsi été possible d'extraire automatiquement des images des pages, les parties représentant les illustrations, afin de les réutiliser dans les documents LaTeX et HTML. Nous avons essayé d'utiliser DSSSL, mais il ne dispose pas de types de données suffisants pour gérer dynamiquement les styles décrits à l'intérieur du document. Bien que le langage sur lequel repose DSSSL (Scheme) permette de définir ces structures de données, le programme résultant est beaucoup trop lent pour être utilisable. De plus, DSSSL ne peut pas accéder à des traitement externes, comme celui d'extraire automatiquement les zones d'illustrations.

6. Conclusion et perspectives

Nous avons atteint un bon degré de similitude, mais il est tout à fait possible de pousser plus avant la description des documents. Que pourrait-on vouloir améliorer ?

Nous avons utilisé un modèle de page simple, composé d'un texte central, un en-tête, un en-pied et une marge. Il existe d'autres modèles de pages, qu'il

serait intéressant de pouvoir décrire. Nous aurions alors besoin d'exprimer ce nouveau modèle dans le document TEI, qui ne connaît aujourd'hui que la notion de texte principal et de notes. Un tel modèle de page pourrait décrire par exemple des journaux ou des livres glosés.

Pour des documents un peu anciens, il est aussi intéressant de garder un répertoire des caractères utilisés pour la composition originelle. Les manuels de typographie célèbres ont été composés avec des polices originales contenant des ligatures ou des graphiques qui ont disparu aujourd'hui. Pour les imprimés anciens qui voulaient ressembler aux manuscrits, un tel répertoire rendrait d'importants services aux philologues.

Pour les documents imprimés, une plus grande précision dans le positionnement des illustrations et des notes serait utile. La limite aujourd'hui est le temps nécessaire pour décrire finement cette présentation. C'est la raison pour laquelle nous nous orientons vers une approche mixte structuration/analyse pour améliorer la qualité de numérisation par l'analyse et la qualité de l'analyse par une pré-numérisation. Il ne nous semble pas raisonnable d'aller au-delà de ce que nous avons fait sur l'encyclopédie de la chose imprimée «à la main».

Malgré ces limitations, nous pensons que notre travail permet de combler une lacune importante dans les bibliothèque électroniques : certains documents textuels à intérêts graphiques peuvent maintenant être décrits assez précisément pour rester utiles.

Un travail important reste encore à faire pour les documents textuels moins réguliers, tels les premiers imprimés ou les manuscrits. Nous pensons qu'il serait dommage de priver les bibliothèques numériques de ces ouvrages, qui restent les plus fragiles et difficiles d'accès.

Bibliographie

[1] Sharon Adler, Anders Berglund, James Clark, Istvan Cseri, Paul Grosso, Jonathan Marsh, Gavin Nicol, Jean Paoli, David Schach, Henry S. Thompson, and Chris Wilson. A Proposal for xsl. `http://www.w3.org/TR/NOTE-XSL.htm`, August 1997.

[2] Martin Bryan. A TEX user's guide to ISO's Document Style Semantics and Specification Language (DSSSL). *TUGboat*, 14(3):223–226, October 1993.

[3] CSS2 Specification — W3C Working Draft. `http://www.w3.org/TR/WD-CSS2/`, Novembre 1997

[4] John Dreyfus et François Richaudeau, editor. *La chose imprimée*. RETZ, 2, rue du Roule - 75001 Paris, 2e édition, 1985.

[5] Pierre-Simon Fournier le jeune. *Manuel typographique*, volume Tome 1. Imprimé par l'auteur, Chez Barbou, rue S. Jacques, 1764.

[6] Almuth Grésillon. *Éléments de critique génétique — lire les manuscrits modernes*. PUF, Paris, 1994.

[7] Karen Murphy. Women Writers Project. http://www.wwp.brown.edu/.

[8] Hélène Richy, Christèle Hérault et Jacques André. Notion de feuille de style. *Cahiers GUTenberg*, no 21:127–134, juin 1995.

[9] C. M. Sperberg-McQueen. Specifying document structure: Differences in LaTeX and TEI markup. *TUGboat*, 12(34):415–421, November 1991.

[10] C. M. Sperberg-McQueen and Lou Burnard, editors, *Guidelines for Electronic Text Encoding and Interchange (TEI P3), Volumes 1 and 2*. The Association for Computers and the Humanities, the Association for Computational Linguistics, and the Association for Literary and Linguistic Computing, Chicago and Oxford, 1994.

[11] Larry Wall and Randal L. Schwartz. *Programming Perl*. O'Reilly & Associates, Inc., 981 Chestnut Street, Newton, MA 02164, USA, 1992. The authoritative guide to `perl`— the programming language for any serious UNIX users.