OpenEdition Press

Qu’est-ce que la Text Encoding Initiative ? | Lou Burnard

La TEI et le XML

Texte intégral

La TEI met l’accent sur ce qui est partagé par tous les types de documents, qu’ils soient représentés

physiquement sous une forme numérique sur un disque ou une carte mémoire, sous une forme imprimée comme un livre ou un journal, sous une forme écrite comme un manuscrit ou un codex, ou sous une forme inscrite dans la pierre ou sur une tablette de cire. Cette continuité facilite la migration du texte depuis des manifestations plus anciennes, comme l'imprimé ou le manuscrit, vers d'autres plus récentes comme le disque ou l'écran. C'est pourquoi la vision de la TEI de ce qu'*est* le texte est largement conditionnée par ce que le texte *a été* dans le passé, sans toutefois trop compromettre ce que le texte peut devenir dans le futur. Elle essaie de traiter tous les types de documents numériques de la même façon, qu'ils soient « nativement numériques » ou non.

2 Par conséquent, la TEI fournit un cadre utile pour penser la nature du texte : elle constitue une sorte d'encyclopédie de notions textuelles admises par le plus grand nombre. Dans ce bref guide nous allons tenter de donner des exemples de quelques-unes de ces notions, en utilisant le vocabulaire défini par la TEI dans ses *Guidelines*.

3 Actuellement, les documents TEI sous forme numérique sont exprimés en utilisant un langage d'encodage formel très largement utilisé appelé XML ou

« *extensible markup language* », publié pour la première fois en 1998 par le World Wide Web Consortium (W3C), mais puisant ses origines dans les systèmes de préparation des documents des années 1980. Le XML offre une manière simple de représenter des données structurées comme un flux linéaire de caractères, et de marquer des parties spécifiques de ce flux avec des « balises » nommées pour indiquer une fonction structurelle ou des éléments de sémantique. Parce que le XML est devenu une technologie extrêmement répandue, nombre d'excellents guides introductifs sont disponibles ailleurs et nous partirons donc du principe que le lecteur a une compréhension basique de concepts-clés tels que ceux d'« élément », d'« attribut », de « schéma », d'« espace de noms », brièvement évoqués ci-dessous par souci d'intelligibilité. Les *Guidelines* de la TEI comportent une « initiation en douceur » au XML qui peut être utile au novice, mais de nombreux autres tutoriels existent sur le sujet.

Voici un exemple de document XML minimaliste :

```
1  <?xml version="1.0"?>
2      <doc xmlns="http://example.o
3          <p n="1">Ceci est un par
4          <p n="2">Ce paragraphe m
```

```
5          </doc>
```

La première ligne d'un document XML prend toujours la forme montrée ci-dessus : une sorte d'instruction spéciale indiquant que ce qui suit est un document XML conforme à la version du standard XML indiquée (dans ce cas, version 1.0). Un document XML consiste en une séquence de caractères lisibles par un humain, sans addition de codes spéciaux ou données binaires. Les caractères < et > sont utilisés pour marquer le début et la fin des *balises* au sein de cette séquence. Une balise peut être ouvrante (comme) ou fermante (comme). Une balise commence toujours par un nom (doc, p, placeName dans l'exemple ci-dessus) et peut aussi contenir des spécifications d'attributs (comme n="1"). L'objet d'une balise ouvrante est de marquer le point dans la séquence de caractères où débute un *élément*, de type indiqué par le nom de la balise, et l'objet de la balise fermante est de marquer la fin de cet élément. L'objet d'une spécification d'attribut est d'ajouter de l'information supplémentaire sur une occurrence d'élément, au-delà de son nom. Dans l'exemple ci-dessus, nous avons un élément nommé <doc> qui contient deux éléments . Les éléments ont tous deux un attribut @n[1] qui donne un chiffre, et tous deux contiennent du texte

brut. Le second élément <p> contient aussi un élément appelé <placeName>.

6 Un document XML comme celui-ci est dit *bien formé* s'il respecte la syntaxe illustrée ici, avec balises ouvrantes et fermantes présentes et correctement imbriquées. Mais le standard XML ne dit rien sur la façon dont les éléments ou attributs doivent être nommés (contrairement au HTML, par exemple, qui définit un ensemble spécifique de balises qui doivent être utilisées d'une façon particulière dans tous les documents), et moins encore sur ce que leur nom signifie. Nous pouvons supposer que les éléments <p> ci-dessus encodent des paragraphes numérotés, mais il n'y a rien dans la représentation XML pour donner raison à cette supposition – ils pourraient tout aussi bien encoder des pages, ou des entrées d'un glossaire, ou des versets. Par conséquent, si l'on trouve un autre document contenant des éléments <p>, comment savoir s'ils ont la même fonction ? La fonction de l'attribut @xmlns ci-dessus est d'aider à résoudre ce problème en fournissant une valeur par défaut pour ce que l'on appelle l'*espace de noms* de tous les éléments contenus par l'élément <doc>.

7 Il n'est pas inhabituel de rencontrer des éléments appartenant à plusieurs espaces de noms dans un

même document : par exemple, un document contenant des notations musicales, des graphiques vectoriels et du texte, tous représentés en XML, peut utiliser des balises de trois espaces de noms différents, un pour les éléments musicaux, un pour les éléments graphiques, et un pour les éléments textuels. Un espace de noms est une façon de marquer un groupe d’éléments : dans notre exemple, son usage fait clairement apparaître que les éléments <p> ici sont différents de tous autres éléments <p> définis par d’autres espaces de noms.

3 On introduit un balisage dans un document pour l’étiqueter et l’organiser en vue d’un traitement automatisé. Si les paragraphes sont clairement marqués, alors un logiciel de mise en forme pourra les mettre en page correctement. Si les noms de lieux sont clairement marqués, un programme peut les sélectionner automatiquement pour générer un index géographique. Mais cela n’est possible de manière fiable que si nous pouvons exercer un certain contrôle sur la façon dont les balises sont introduites dans le document et l’endroit où elles apparaissent. La technologie XML fournit ce niveau additionnel de contrôle grâce à ce que l’on appelle un *schéma*, une sorte de combinaison de lexique et de grammaire pour les documents XML valides. Nous avons noté plus haut qu’un document

XML est dit bien formé s'il respecte les règles syntaxiques du standard XML. Il peut aussi, de manière optionnelle, être dit *valide* si les balises qu'il contient se conforment à un schéma.

Un *schéma* spécifie un ensemble de noms d'éléments, les noms et le type de données de tous les attributs qui leur sont associés, et les règles relatives aux contextes dans lesquels ils peuvent apparaître. Un schéma pour notre exemple simple ci-dessus dira qu'il existe des éléments nommés <doc>, <p>, <placeName>, etc. Il pourrait aussi spécifier que les éléments <p> peuvent apparaître à l'intérieur des éléments <doc>, que les <placeName> peuvent apparaître à l'intérieur des <p>, que l'attribut @n doit avoir une valeur numérique, etc. Notez cependant qu'un schéma XML n'a toujours pas de moyen de spécifier que la balise <placeName> indique le nom d'un lieu, ou ce que nous entendons par *lieu* : de telles contraintes sémantiques additionnelles doivent être spécifiées ailleurs, par exemple dans une documentation telle qu'en fournissent les *Guidelines* de la TEI.

La TEI fournit le nom et la définition de centaines de balises, en même temps que des règles sur la façon dont elles peuvent être combinées. Plus précisément, les *Guidelines* de la TEI définissent cinq ou six cents

concepts différents, avec les spécifications détaillées des éléments et classes d'éléments XML qui peuvent être utilisés pour les représenter. La plupart des documents TEI, si ce n'est la totalité, n'a besoin que d'une petite partie de ce qui est fourni. C'est pourquoi il est trompeur de penser la TEI comme un schéma monolithique. Pour faciliter l'interopérabilité, chaque document TEI utilise des composants empruntés au même gigantesque schéma, mais la plupart des projets TEI utilisent des sous-ensembles très restreints, et un projet bien organisé a généralement sa propre documentation personnalisée identifiant ce sous-ensemble.

1 Des logiciels tels que l'application Web Roma peuvent être utilisés pour faire une sélection dans les spécifications de la TEI et générer à partir de cela un schéma approprié aux besoins de votre propre projet, ou bien vous pouvez simplement assembler un schéma manuellement. Nous aborderons ce sujet dans de plus amples détails dans un prochain chapitre. Les *Guidelines* de la TEI, librement accessibles depuis le site web http://www.tei-c.org.proxy.bib.ucl.ac.be:8888/Guidelines, constituent un manuel de référence complet pour ces concepts, combinant spécifications techniques et discussions

détaillées sur leur usage.

12 Il y a un grand nombre d'outils logiciels disponibles pour créer, transformer et traiter des documents XML en général, ou XML TEI en particulier. C'est un sujet très vaste aux évolutions rapides qui n'est pas traité ici, mais vous trouverez quelques liens utiles sur le site web de la TEI et sur son Wiki.

Notes

1. Par convention, les noms d'attributs mentionnés dans le texte sont précédés du signe @.

© OpenEdition Press, 2015

Creative Commons - Attribution-NonCommercial-NoDerivs 3.0 Unported - CC BY-NC-ND 3.0

Référence électronique du chapitre

BURNARD, Lou. *La TEI et le XML* In : *Qu'est-ce que la Text Encoding Initiative ?* [en ligne]. Marseille : OpenEdition Press, 2015 (généré le 29 mai 2017). Disponible sur Internet : <http://books.openedition.org.proxy.bib.ucl.ac.be:8888/oep/1298>. ISBN : 9782821855816. DOI : 10.4000/books.oep.1298.

Référence électronique du livre

BURNARD, Lou. *Qu'est-ce que la Text Encoding Initiative ?* Nouvelle édition [en ligne]. Marseille : OpenEdition Press, 2015

(généré le 29 mai 2017). Disponible sur Internet : <http://books.openedition.org.proxy.bib.ucl.ac.be:8888/oep/1237>. ISBN : 9782821855816. DOI : 10.4000/books.oep.1237.
Compatible avec Zotero