

Formation Edition numérique

L'ODD (*One Document Does it all*)

Simon Gabay



Personnalisier

Ma TEI

La TEI doit (théoriquement) apporter une solution à toutes les situations possibles: l'éventail est donc logiquement trop large pour un projet personnel, même ambitieux. Il faut donc

- Restreindre la TEI à ses besoins en concevant un modèle adapté à ses besoins
- Documenter ce modèle pour expliciter ses choix aux autres, afin de les rendre lisibles et réexploitables

Des exemples de documentation

Les projets proposent leur documentation

- *Bibliothèques virtuelles humanistes*: <http://www.bvh.univ-tours.fr/XML-TEI/index.asp>
- *Ecole des chartes*:
<http://developpements.enc.sorbonne.fr/diple/schema/>
- *ENRICH*: <http://projects.oucs.ox.ac.uk/ENRICH/>

Typologie des modification

- Suppression d'éléments
- Changement du noms d'éléments
- Modifications d'un modèle de contenu
- Modification d'une liste d'attributs ou de valeurs d'attributs, d'un type de contenu
- Modification de l'appartenance à une classe
- Ajout de nouveaux éléments

TEI conformance

Deux règles importantes :

1. Une personnalisation est une restriction de la TEI (elle reste valide contre *TEI All*)
2. Elle ne doit pas altérer le sémantisme des éléments

Si la personnalisation respecte ces deux règles, elle est dite "propre" (*clean modification*), sinon elle est dite "sale" (*unclean modification*). Une modification sale peut cependant être nécessaire.

Typologie des changements

Changements propres:

- Suppression d'éléments
- Modifications d'un modèle de contenu
- Modification d'une liste d'attributs ou de valeurs d'attributs, d'un type de contenu
- Modification de l'appartenance à une classe

Changement sale:

- Changement du nom d'éléments

Changement sale ou propre:

- Ajout de nouveaux éléments (propre si ces nouveaux éléments appartiennent à un autre espace de nom).

Solutions simples

Les customisations:

- *All* : toute la TEI
- *Drama*: pour le théâtre
- *Corpus*: pour l'encodage de corpus linguistique
- *Lite*: version réduite de la TEI
- *Bare*: version minimale de la TEI

La personnalisation

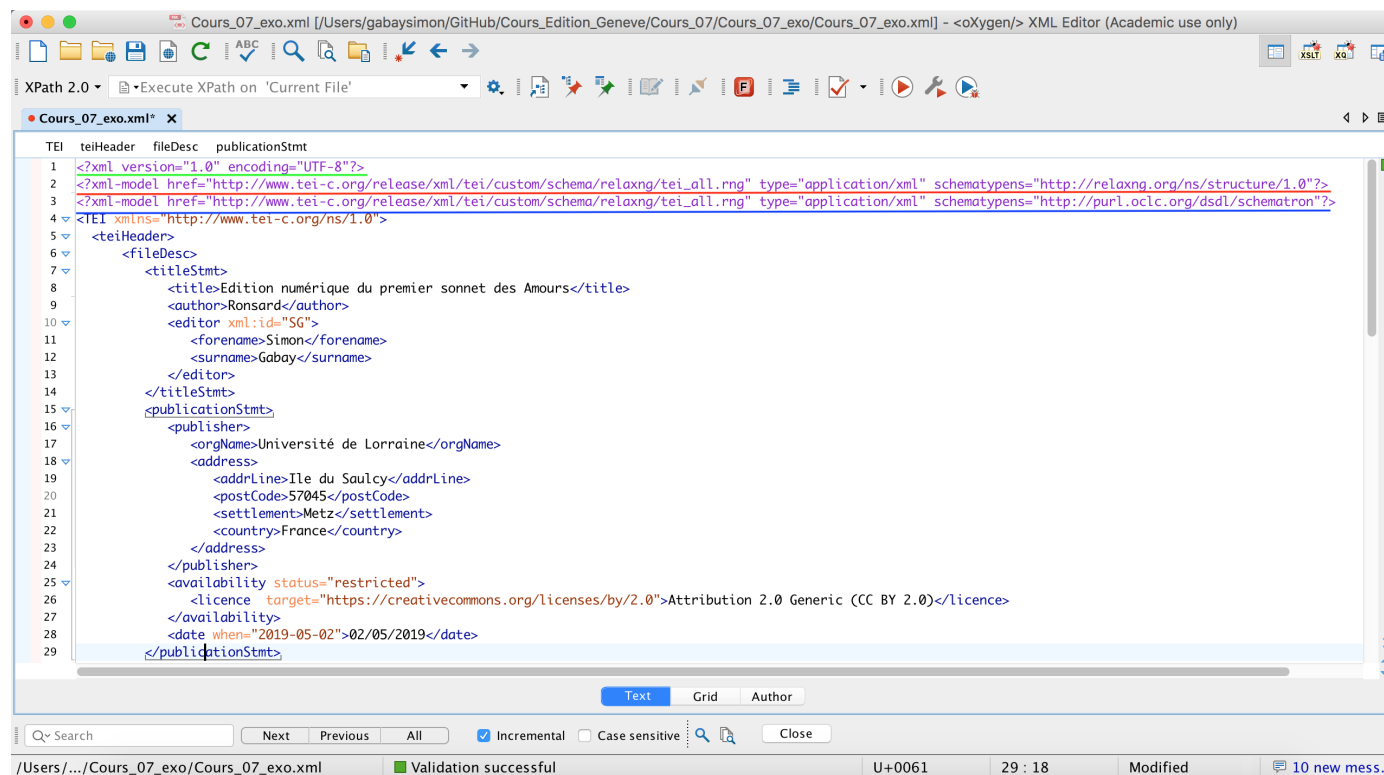
- *oddByExample*
- *Roma*

Un Schema

Construction

Les fichiers TEI commencent par trois informations:

- La déclaration XML (Vert)
- Le lien vers les schema (rouge) et les règles schématron (bleu)



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei_all.rng" type="application/xml" schematypens="http://relaxng.org/ns/structure/1.0"?>
3 <?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei_all.rng" type="application/xml" schematypens="http://purl.oclc.org/dsdl/schematron"?>
4 <TEI xmlns="http://www.tei-c.org/ns/1.0">
5   <teiHeader>
6     <fileDesc>
7       <titleStmt>
8         <title>Edition numérique du premier sonnet des Amours</title>
9         <author>Ronsard</author>
10        <editor xml:id="SG">
11          <forename>Simon</forename>
12          <surname>Gabay</surname>
13        </editor>
14      </titleStmt>
15      <publicationStmt>
16        <publisher>
17          <orgName>Université de Lorraine</orgName>
18          <address>
19            <addrLine>Ile du Saulcy</addrLine>
20            <postCode>57045</postCode>
21            <settlement>Metz</settlement>
22            <country>France</country>
23          </address>
24        </publisher>
25        <availability status="restricted">
26          <licence target="https://creativecommons.org/licenses/by/2.0">Attribution 2.0 Generic (CC BY 2.0)</licence>
27        </availability>
28        <date when="2019-05-02">02/05/2019</date>
29      </publicationStmt>
```

Un nouveau schéma

Notre nouveau schéma est une restriction de la TEI qui reste valide contre *TEI All*. Il existe plusieurs formats:

- XML Schema (`.xsd`) [[cf. wikipedia](#)]
- RELAX NG XML (`.rng`) [[cf. wikipedia](#)]
- RELAX NG Compact (`.rnc`)

Le dernier est le plus populaire parmi les utilisateurs de la TEI.

RELAX NG

Relax NG est écrit en XML et est donc à peu près lisible. La règle autorisant cet encodage:

```
<lg>  
  <l>C'est un trou de verdure où chante une rivière</l>  
  <l>Accrochant follement aux herbes des haillons</l>  
</lg>
```

est la suivante (dans une version très minimale)

```
<element name="lg">  
  <oneOrMore>  
    <element name="l">  
      <text/>  
    </element>  
  </oneOrMore>  
</element>
```

RELAX NG Compact

C'est une version réduite de *RELAX NG* qui n'est pas écrite en XML.

Pour le même texte

```
<lg>  
  <l>C'est un trou de verdure où chante une rivière</l>  
  <l>Accrochant follement aux herbes des haillons</l>  
</lg>
```

La règle est désormais la suivante:

```
element book {  
  element page { text }+  
}
```

Il est très simple de passer de la version compact à la version XML de *RELAX NG* et inversement.

ODD

Pourquoi l'ODD?

Les règles des schémas sont complexes à écrire et nécessitent une très bonne maîtrise du code. L'ODD sert à contourner ce problème.

- L'*ODD* est un fichier TEI normal, constitué d'un `<teiHeader>` et d'un `<body>`. Si je maîtrise déjà la TEI, je comprends facilement son fonctionnement.
- L'*ODD* permet de générer le schéma, mais aussi la documentation du projet en HTML
- L'*ODD* est accompagné d'une gamme d'outil qui permettent d'effectuer les transformations vers tous les formats nécessaires.

Un double `<body>`

S'il n'y a bien qu'un seul `<body>` dans l'*ODD*, on peut distinguer deux temps:

1. Une série de `<div>` où j'explique mes choix en rédigeant du texte.
2. Une partie plus technique et plus rigide où je précise la constitution de mes classes d'attributs (`<specGrp>`) et surtout des modules pour les éléments (`<schemaSpec>`).

```
<body>
  <div>
    <p>J'ai fait comme-ci, comme-ça.</p>
  </div>
  <div>
    <specGrp/>
  </div>
  <div>
    <schemaSpec/>
  </div>
</body>
```


Les modules

Concentrons-nous sur `schemaSpec` : il s'agit d'autoriser des éléments: comment les sélectionner? La TEI a en fait regroupés chaque élément dans une module. Tous sont amplement commentés dans les *Guidelines*

Module name	Formal public identifier	Where defined
analysis	Analysis and Interpretation	17 Simple Analytic Mechanisms
certainty	Certainty and Uncertainty	21 Certainty, Precision, and Responsibility
core	Common Core	3 Elements Available in All TEI Documents
corpus	Metadata for Language Corpora	15 Language Corpora
dictionaries	Print Dictionaries	9 Dictionaries
drama	Performance Texts	7 Performance Texts
figures	Tables, Formulae, Figures	14 Tables, Formulae, Graphics and Notated Music
gaiji	Character and Glyph Documentation	5 Characters, Glyphs, and Writing Modes
header	Common Metadata	2 The TEI Header
iso-fs	Feature Structures	18 Feature Structures
linking	Linking, Segmentation, and Alignment	16 Linking, Segmentation, and Alignment
msdescription	Manuscript Description	10 Manuscript Description
namesdates	Names, Dates, People, and Places	13 Names, Dates, People, and Places
nets	Graphs, Networks, and Trees	19 Graphs, Networks, and Trees
spoken	Transcribed Speech	8 Transcriptions of Speech
tagdocs	Documentation Elements	22 Documentation Elements
tei	TEI Infrastructure	1 The TEI Infrastructure
textcrit	Text Criticism	12 Critical Apparatus
textstructure	Default Text Structure	4 Default Text Structure
transcr	Transcription of Primary Sources	11 Representation of Primary Sources
verse	Verse	6 Verse

On retrouve le nom du module dans les *guidelines* (et, surtout, après quelques années on les connaît par cœur).

<lg>

(groupe de vers) contient un groupe de vers fonctionnant comme une unité formelle, par exemple une strophe, un refrain, un paragraphe en vers, etc. [3.12.1 Core Tags for Verse 3.12 Passages of Verse or Drama 7.2.5 Speech Contents]

Module	core — Elements Available in All TEI Documents
Attributs	att.global (@xml:id, @n, @xml:lang, @xml:base, @xml:space) (att.global.rendition (@rend, @style, @rendition)) (att.global.linking (@corresp, @synch, @sameAs, @copyOf, @next, @prev, @exclude, @select)) (att.global.analytic (@ana)) (att.global.facs (@facs)) (att.global.change (@change)) (att.global.responsibility (@cert, @resp)) (att.global.source (@source)) att.divLike (@org, @sample) (att.metrical (@met, @real, @rhyme)) (att.fragmentable (@part)) att.typed (@type, @subtype) att.declaring (@decls)
Membre du	macro.paraContent model.divPart
Contenu dans	core: add corr del emph head hi item lg note orig p q quote ref reg said sic sp stage title unclear dictionaries: case colloc def gen gram hyph iType lang lbl mood number orth per pos pron stress subc syll tns usg drama: camera caption castList epilogue performance prologue set sound tech view figures: cell figure

moduleRef

Pour autoriser un élément, je dois ajouter son module d'abord, puis les éléments du modules que je choisis de conserver (je choisir la logique inverse, et en exclure)

```
<schemaSpec ident="monProjet" docLang="fr">
  <moduleRef key="textstructure"
             include="TEI text body div front back"/>
  <moduleRef key="namesdates"
             include="forename surname orgName persName"/>
</schemaSpec>
```

Certains modules sont obligatoires, comme `core` ou `tei`.

elementSpec : les éléments contenus

Je peux ensuite préciser le comportement de mon élément avec `elementSpec`. Je peux préciser les autres éléments qu'il peut contenir

```
<elementSpec ident="monElement" mode="change">
  <content>
    <elementRef key="element1"/>
    <elementRef key="element2" minOccurs="1"
maxOccurs="unbounded"/>
    <sequence>
      <elementRef key="element3"/>
      <elementRef key="element4"/>
    </sequence>
  </content>
</elementSpec>
```

elementSpec : les attributs contenus

Je peux ensuite préciser le comportement de mon élément avec `elementSpec`. Je peux préciser les autres éléments qu'il peut contenir

```
<elementSpec ident="monElement" mode="change">
  <attList>
    <attDef ident="att1" mode="delete"/>
    <attDef ident="att2" mode="change" usage="req">
      <datatype>
        <dataRef key="teidata.count"/>
      </datatype>
    </attDef>
    <attDef ident="att3" mode="change" usage="req">
      <valList mode="add" type="closed">
        <valItem ident="valeur1"/>
        <valItem ident="valeur2"/>
      </valList>
    </attDef>
  </attList>
</elementSpec>
```

Les classes

En plus des modules, les éléments sont regroupés par classes. Tous les éléments d'une même classe ont (plus ou moins) le même comportement:

- Ils apparaissent au même endroit
- Ont les mêmes attributs
- ...

On peut les utiliser pour changer tous les éléments d'une classe d'un coup, notamment pour restreindre la liste d'attributs.

On retrouve le nom des classes d'attributs dans les *guidelines* (et, surtout, après quelques années on les connaît par cœur).

 <Text Encoding Initiative>

Home Guidelines Activities Tools Membership Support About News

P5 Guidelines — Français Search

P5: Recommandations pour l'encodage et l'échange de textes électroniques

Version 4.0.0. Last updated on 13th February 2020, revision ccd19b0ba

<lg>

Accueil
C Éléments

<lg> (groupe de vers) contient un groupe de vers fonctionnant comme une unité formelle, par exemple une strophe, un refrain, un paragraphe en vers, etc. [3.12.1 Core Tags for Verse 3.12 Passages of Verse or Drama 7.2.5 Speech Contents]

Module	core — Elements Available in All TEI Documents
Attributs	att.global (@xml:id, @n, @xml:lang, @xml:base, @xml:space) (att.global.rendition (@rend, @style, @rendition)) (att.global.linking (@corresp, @synch, @sameAs, @copyOf, @next, @prev, @exclude, @select)) (att.global.analytic (@ana)) (att.global.facs (@facs)) (att.global.change (@change)) (att.global.responsibility (@cert, @resp)) (att.global.source (@source)) att.divLike (@org, @sample) (att.metrical (@met, @real, @rhyme)) (att.fragmentable (@part)) att.typed (@type, @subtype) att.declaring (@decls)
Membre du	macro paraContent model divPart
Contenu dans	core: add corr del emph head hi item lg note orig p q quote ref reg said sic sp stage title unclear dictionaries: case colloc def gen gram hyph iType lang lbi mood number orth per pos pron stress subc syll tns usg drama: camera caption castList epilogue performance prologue set sound tech view figures: cell figure

specGrp

Je vais pouvoir modifier les classes avec specGrp

```
<specGrp xml:id="classmods">
  <classSpec ident="att.global" mode="change"
             module="tei" type="atts">
    <attList>
      <attDef ident="xml:space" mode="delete"/>
      <attDef ident="xml:base" mode="delete"/>
    </attList>
  </classSpec>
  <classSpec ident="att.divLike" mode="delete"
             module="tei" type="atts"/>
</specGrp>
```