

TEI – Travaux pratiques : ODD

JB Camps – M2 HN & TNAH – 10 janvier 2018

Personnaliser la TEI (1/2)

Le modèle de la TEI

1. Quel est le nom du module permettant la réalisation de notices de manuscrits, et quel est le chapitre qui lui est consacré ?
2. Quel est le module permettant de réaliser des fichiers ODD ? où est-il documenté ?
3. Quel module ai-je besoin d'inclure dans ma personnalisation pour pouvoir utiliser l'élément `gap` ?
4. Quelles sont les classes membres d'`att.written` ?
5. À quel(s) attribut(s) aurait accès un élément membre d'`att.written` ? Membre d'`att.textCritical` ?
6. Si je fais référence à `model.global.edit`, à quel(s) ensemble(s) d'éléments fais-je référence ?
7. Quelles sont les classes utilisant le `model.respLike` ? Quel(s) élément(s) l'utilise(nt) ? Quels éléments recouvre-t-il ?
8. Quel contenu recouvre la `macro.phraseSeq` ? Donner un exemple d'élément l'utilisant.
9. Quel est le contenu possible pour l'attribut `@ana` ? Quelle macro est-elle utilisée ?

Utilisation de *Roma*

10. Aller sur <http://www.tei-c.org/Roma/>, choisir de construire à partir du plus petit modèle possible ;
11. Renseigner les informations sur le modèle ;
12. Choisir la langue de la documentation ;
13. Sélectionner les modules qui pourraient vous être utiles ;
14. Dans ces modules, exclure les éléments qui ne sont pas nécessaires ;
15. Redéfinir l'attribut `@type` de `note`, pour le rendre obligatoire et lui donner une liste fermée de valeurs possibles ;
16. Modifier ou supprimer les classes d'attribut du modèle ; *a minima*, redéfinir `att.global.rendition` pour en retirer les attributs qui ne paraissent pas pertinents.
17. Exporter ce modèle sous forme de document ODD, de documentation HTML et de schéma RelaxNG.
18. Regarder le fichier ODD qui en résulte ; observer notamment les effets de votre sélection de modules, d'éléments, de vos redéfinitions d'attributs et de classes.

Utilisation de *Oddbyexample*

19. Installer, si ce n'est déjà fait, le module complémentaire TEI d'`<oXygen/>` ;
20. Configurer le scénario de transformation *oddbyexample* ;
21. Placer dans un dossier les documents encodés à partir desquels on souhaite réaliser le modèle ;
22. L'appliquer aux documents TEI que vous avez encodés ;
23. Regarder le résultat.

Découverte de la syntaxe ODD

24. Lire les deux modèles ODD créés au cours de la séance, et tenter de comprendre leur syntaxe.

Personnaliser la TEI : la syntaxe ODD (2/2)

Structure générale

25. Observer la structure générale du fichier `tei-cartes.odd`, ainsi que du fichier ODD que vous avez généré. Vérifier, dans ce dernier, que `@docLang` et `@xml:lang` soient bien identiques et égaux à `'fr'`.

Spécifications techniques

Modules

26. Observer la définition du module `textcrit` des *Guidelines* de la TEI, et des modules de la MEI.
27. Modifier les références aux modules dans votre fichier ODD, pour exclure les éléments que vous ne souhaitez pas importer, ou inclure seulement ceux que vous souhaitez importer.
28. Convertir votre fichier ODD en xhtml, via le scénario existant, et observer les changements que cela engendre.

Éléments

29. Observer la définition de l'élément `supplied` dans les sources des *Guidelines*.
30. Redéfinir l'élément `supplied`, pour supprimer son appartenance à la classe `model.choicePart`.
31. Transformer le fichier ODD en xhtml, et observer les différences que cela induit pour cette classe, pour l'élément `supplied`, pour l'élément `choice`.
32. Redéfinir le contenu de l'élément `supplied`, pour le restreindre à du seul texte.
33. Redéfinir l'attribut `@reason` de `supplied`, pour le rendre obligatoire, pour redéfinir son type de données (`datatype`) en une seule occurrence de `teidata.word`, et ajouter une liste semi-fermée de valeurs possibles (`'over-binding'`, `'faded-ink'`, `'damage'`), que l'on documentera. Modifier, si nécessaire, la description de cet attribut pour la faire concorder avec ses nouvelles spécifications.
34. Modifier l'élément `app` (`'change'`), pour que son contenu soit constitué dans cet ordre (`@preserveOrder`) et sans contenu mixte, d'un et un seul élément `lem` (obligatoire) suivi d'au moins un élément `rdg`, et de `witDetail` optionnels.
35. Modifier l'élément `msDesc`, pour que son contenu se compose d'un élément `msIdentifier` obligatoire, d'un titre (`model.headLike`) obligatoire, puis de, soit un paragraphe de texte, soit les éléments `msContents`, `physDesc`, `history` (obligatoires), `additionnal` (optionnel et non répétable) et `msFrag` (optionnel et répétable).
36. Redéfinir localement, au niveau de `msDesc`, `@type`, pour le rendre obligatoire et lui donner deux seules valeurs possibles : `'short'` (notice courte) ou `'full'` (notice complète), et en les documentant.
37. Ajouter une remarque en français à l'élément `msDesc` pour expliquer notre parti-pris.
38. Ajouter une contrainte, exprimée en ISO-Schematron, permettant de s'assurer que l'emploi d'`@type` concorde avec les éléments employés.

Classes

39. Lire la spécification des classes `model.pLike` et `att.global`.
40. Supprimer la classe `att.dataable.iso` et inspecter votre modèle pour voir s'il contient également des classes inutilisées.
41. Modifier la définition d'`att.global`, pour supprimer `@xml:base` et `@xml:space`.

Macros et type de données

42. Lire la spécification de `teidata.certainty`. Le redéfinir pour supprimer la valeur `'unknown'`. Documenter.

Documentation

43. Observer la structure d'un chapitre des *Guidelines* (dossier `Guidelines/en/`).
44. Créer une `div`, d'`@xml:id` = `'transcr'` et avec un titre approprié (`head`), dans laquelle on expliquera les principes de transcription, et, notamment, l'utilisation de `supplied`. Ajouter des exemples avec `egXML`.
45. Créer une `div`, d'`@xml:id` = `'msdescription'` et avec un titre approprié (`head`), dans laquelle on expliquera, avec des paragraphes de prose, et en faisant référence aux spécifications, l'emplacement attendu pour les notices, les types de notice et les recommandations aux encodeurs.

Mise en œuvre

46. Créer un nouveau document XML, en l'associant au schéma.
47. Y insérer une notice de manuscrit, ajuster son `@type`, ... ; utiliser `supplied` ou `app`, constater les changements.