

INU3011 Documents structurés

Cours 7 Modélisation 2

Plan

- Retour sur le Quiz 2
- Retour sur Cours 6
- Présentation du TP 4
- Quelques mots sur les TP 5 et 6
- Modélisation : points spécifiques
- Regard sur vins

Quiz 2

- Moyenne $24,51 / 30 = 81,7 \%$
- Question "ajout de commentaires"

Retour sur Cours 6

Principes (rappel)

1. Modélisation frugale
2. Découpage adéquat
3. Balisage logique
4. Regroupements logiques
5. Restriction maximale
6. Reflet des contraintes sémantiques

Exemple 3 de contrainte sémantique validable par modèle de contenu (1/4)

- Choix avec "autre, préciser" :

Moyen de transport :

Train

Avion

Autre, préciser : _____

Exemple 3 de contrainte sémantique validable par modèle de contenu (2/4)

- Première modélisation :

```
<!ELEMENT transport  
    ((train | avion | autre), préciser?)>  
<!ELEMENT train EMPTY>  
<!ELEMENT avion EMPTY>  
<!ELEMENT autre EMPTY>  
<!ELEMENT préciser (#PCDATA)>
```

Exemple 3 de contrainte sémantique validable par modèle de contenu (3/4)

- Cette modélisation est trop permissive
- Permet :

<transport><autre/>

<préciser>bateau</préciser></transport>

mais aussi :

<transport><train/>

<préciser>bateau</préciser></transport>

qui, sémantiquement, n'a pas de sens

Exemple 3 de contrainte sémantique validable par modèle de contenu (4/4)

- Meilleure modélisation :

```
<!ELEMENT transport  
    (train | avion | autre-préciser)>  
<!ELEMENT train EMPTY>  
<!ELEMENT avion EMPTY>  
<!ELEMENT autre-préciser (#PCDATA)>
```

TP 4

Modélisation préliminaire

Travail en équipe (15%)

TP 5 et 6

Modélisation

Points spécifiques (1/2)

- Bonnes pratiques pour les modèles de contenu
- Approximation de contenus textuels contrôlés
- Attribut ou sous-élément?
- Éléments multimédia

Modélisation

Points spécifiques (2/2)

- Liens Web
- Liens entre documents
- Information "binaire" (oui/non)

Bonnes pratiques pour les modèles de contenu

Découlent des principes de modélisation

Modèle de contenu constitué d'un nom d'élément seul (1/3)

- Un nom d'élément (défini ailleurs dans la DTD) est *techniquement possible* comme modèle de contenu d'un autre élément
- Cette situation doit cependant être évitée, puisqu'elle amène un niveau hiérarchique inutile...

Modèle de contenu constitué d'un nom d'élément seul (2/3)

- Exemple:
 - <!ELEMENT adresse-perm (adresse)>
 - <!ELEMENT adresse-temp (adresse)>
 - L'élément supérieur indique le rôle de l'objet
 - L'élément inférieur indique le type d'objet

Modèle de contenu constitué d'un nom d'élément seul (3/3)

- On élimine un des 2 éléments (en fusionnant les noms, si nécessaire), ex.:

<!ELEMENT adresse-perm (...)>

<!ELEMENT adresse-temp (...)>

où ... est le modèle de contenu originel de "adresse"

- Autre exemple :

<!ELEMENT auteur (nom)>

<!ELEMENT nom (#PCDATA)>

devient: <!ELEMENT auteur-nom (#PCDATA)>

ou bien: <!ELEMENT nom-de-l-auteur (#PCDATA)>

Modèle de contenu unitaire

- On appelle modèle de contenu unitaire:
 - Un nom d'élément
 - Un choix de noms d'élément
- "Unitaire" parce que tous les contenus acceptés par un tel modèle consistent toujours en exactement un élément
- Tous les autres: *non unitaires*

Particules d'occurrence

- +, * et ? peuvent s'appliquer à autre chose qu'un modèle de contenu unitaire, mais c'est inhabituel *et déconseillé*:
 - Les modèles non unitaires devraient être d'abord placés dans un « élément chapeautant », auquel on appliquera la particule d'occurrence
 - Ex.: ~~(mois, jour)+~~ ⇒ (date+) avec `<!ELEMENT date (mois, jour)>`

Particules d'occurrence (suite)

- De plus, les formes

(modèle-unitaire+)

(modèle-unitaire*)

devraient survenir seulement comme modèles de contenu complets d'un élément, alors que

(modèle-unitaire?)

devrait survenir seulement comme *partie* d'un modèle de contenu d'élément

Particules d'occurrence (suite)

- Exemple: ~~(auteur, date+, sujet)~~ ⇒
(auteur, dates, sujet) avec
<!ELEMENT dates (date+)>
- Exemple: ~~(auteur, date, sujet)~~ avec
~~<!ELEMENT date (modif?)>~~ ⇒
(auteur, date-modif?, sujet)

« Approximation » de contenus textuels contrôlés (1/3)

- Une des limites importantes des DTD :
 - Aucune façon de contrôler ce que l'autrice met dans un élément #PCDATA
 - Ça peut être vide
 - Aucune limite de longueur
 - Aucune façon de limiter à une forme particulière ou à une liste de valeurs possibles

« Approximation » de contenus textuels contrôlés (2/3)

- Cette limite explique en partie le succès d'autres langages de contraintes XML
 - Par ex.: schémas XML du W3C, RelaxNG
- Mais peut-on se rapprocher d'un contrôle de contenu textuel avec les DTD ?

« Approximation » de contenus textuels contrôlés (3/3)

- Oui !
 - Éléments vides avec « | » (OR)
 - Attributs à valeurs énumérées
- Exemples: [160-Ex-contrôle-contenu](#)

Attribut ou sous-élément?

- Parfois, une même information pourrait *a priori* être véhiculée aussi bien par un sous-élément que par un attribut d'un élément existant; comment choisir?
- Texte: [Attributs et modélisation en XML](#)
 - Avec dossier d'exemples :
 - [170-Ex-attribut-vs-sous-element](#)

Éléments multimédias (1/3)

- On parle de fichiers qui existent de façon distincte du document (JPEG, MP3, etc.)
- En principe, ces éléments pourraient être "codés" en texte et intégrés au XML
 - Base64, URL "data:", etc.
 - Exige que le document soit "assemblé" à partir de parties distinctes (texte, image, etc.)
 - Complexifie la saisie

Éléments multimédias (2/3)

- En pratique, on garde l'élément multimédia dans un fichier **séparé** du document
 - Analogue aux images en HTML
 - Grâce au stylage, l'élément pourra être intégré adéquatement au document **au moment de la lecture ou de la publication**
 - Il suffit de prévoir un élément (ou attribut) dont le contenu permettra de reconstituer le **nom** (ou l'URL) du fichier séparé

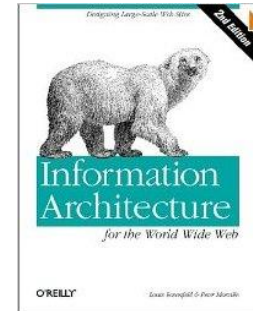
Éléments multimédias (3/3)

- Par exemple, si notre modèle prévoit un élément (ou attribut) qui est un identifiant qui varie d'un document à l'autre
 - Il suffit d'utiliser cet identifiant dans le nom du fichier multimédia séparé
 - Le nom (ou URL) du fichier peut être reconstitué au stylage à partir de l'identifiant qu'on trouve dans le document XML
- Facilement généralisable aux cas de plusieurs éléments multimédias par document

Éléments multimédia: exemples

Le document XML :

```
<livre isbn='0-596-00035-9'>  
...  
</livre>
```



Le fichier image : 0-596-00035-9.jpg

Son URL relative : images/0-596-00035-9.jpg

Au stylage, la bonne URL est placée dans l'extrant HTML en insérant la valeur de l'attribut `isbn` entre les chaînes "images/" et ".jpg":

```

```

[180-Ex-multimedia](#) : fichier XML et extrant HTML (faire afficher la source)

[185-Ex-multimedia-facultatif](#) : possibilité d'élément multimédia manquant

Liens Web

- Les liens Web sont modélisés exactement comme les liens en HTML :
`texte cliquable`
- Mais avec des noms d'élément et d'attribut qui s'inscrivent dans la logique du modèle
 - Ex.: `<lienWeb url="...">texte cliquable</lienWeb>`
- Ces liens seront facilement transformés en liens Web par la feuille de style XSLT

Liens entre documents

- S'assurer que chaque document possède un "identifiant" unique*
- Prévoir un élément (e.g. <lien>) qui contiendra (directement ou dans un attribut) l'identifiant du document-cible
- L'élément <lien> sera facilement transformé en vrai lien HTML par la feuille de style XSLT

*Cet identifiant pourra servir aussi pour l'élément multimédia

Exemples de liens

- Exemples dans le [projet "vins"](#) :
 - [Document XML](#)
 - Faites afficher source, puis...
 - Repérez <lien> et <lien-ext> vers la fin de la source
 - [Document après stylage](#)
 - Faites afficher source; vous verrez le HTML produit par la feuille de style XSLT *directement dans le navigateur Web !*
 - Voyez vers la fin du document en quoi les liens ont été transformés

Informations binaires (1/4)

- Les informations qui reviennent essentiellement à une case à cocher peuvent être modélisées par :
 - Un élément vide facultatif
 - La présence ou absence de l'élément dans le document suffit à véhiculer l'information
 - Le nom de l'élément vide correspond à l'intitulé de la "case à cocher"...
- Ou par un attribut à 2 valeurs ou facultatif

Informations binaires (2/4)

Exemples

- ❑ Confidentiel `<confidentiel />`
- ❑ Version finale `<version-finale />`
- ❑ Urgent `<urgent />`
- ❑ J'accepte qu'on me recontacte plus tard `<accepte-contact-futur />`

Informations binaires (3/4)

Exemple avec attribut à deux valeurs :

Dans la DTD :

```
<!ATTLIST mémo confidentiel (oui | non) #REQUIRED>
```

Dans le document :

```
<mémo confidentiel="oui">...</mémo>
```

ou bien :

```
<mémo confidentiel="non">...</mémo>
```

Informations binaires (4/4)

Exemple avec attribut facultatif :

Dans la DTD :

```
<!ATTLIST mémo confidentiel (oui) #IMPLIED>
```

Dans le document :

```
<mémo confidentiel="oui">...</mémo>
```

ou bien :

```
<mémo>...</mémo>
```

StudiUM

- Exercices de transcription en bien formé
 - C'est-à-dire : rédaction d'*hypothèses de documents valides*
- TP 4
- Semaine prochaine : activités libres