

# INFO-H-509 : Technologies XML

## TP 2 - Langages de schémas pour XML

Professeur : Stijn Vansummeren  
Assistant : François Picalausa  
<http://cs.ulb.ac.be/public/teaching/infoh509>  
Année académique 2009-2010

---

### XML Schema Validity

Le données de ce TP sont disponibles en téléchargement sur le site des TPs.

#### Exercice 1.1

Le document `schedule.xml` décrit l'horaire du cours. Bien que ce document suive les règles de *Well-Formedness*, il ne correspond pas au schéma `schedule.xsd`. Proposer des corrections pour que ce document devienne valide en rapport du schéma.

#### Exercice 1.2

Le schéma est peu exigeant en ce qui concerne le contenu de l'élément `course`. Les contraintes souhaitées sont les suivantes :

Chaque événement (cours ou TP) est associé a un attribut `on` indiquant la date de celui-ci et optionnellement un nom unique, définit par l'attribut `id`. Un événement est aussi associé à un local.

Un événement décrit soit un cours (`lecture`) soit un TP (`exercices`). A la fois les cours et les TPs portent sur des sujets, décrits par des éléments `topic`. Les différents sujets d'un cours sont des uri, donnant des explications détaillées. Les exercices auront eux de un à trois cours pour sujet.

Procéder comme suit pour compléter le schéma :

- Compléter la définition de `eventType`, qui servira de base pour décrire cours et TPs. Il contiendra donc les définitions communes.
- Étendre ce type séparément pour les cours et pour les TPs.
- Compléter la définition de l'élément `course`. Les cours et séances d'exercices peuvent être entrelacés.

#### Exercice 1.3

**Exercice supplémentaire** : Utiliser les éléments `unique`, `key` et `keyref` pour assurer que les ressources d'un cours sont unique, qu'un cours réfère à un local existant et qu'une séance d'exercices a pour sujet un cours existant. Corriger le document si nécessaire.

#### Exercice 1.4

Compléter le schéma suivant pour utiliser les namespaces, comme le fait le document `instance`.

```
<message on="2010-03-23T17:55:00" xmlns="http://www.example.org/message">
  Hello!
</message>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="message">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="on" type="xs:dateTime" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:element name="message" type="m:message" />
</xs:schema>
```

## Expressions régulières et DTD

### Exercice 2.1

Nous considérons les expressions régulières suivantes sur l'alphabet  $a, b, c, d$  :

1.  $a(bb?)?(c^+(a|d))$
2.  $a^+bbccd$
3.  $c?b?(a^*|b|c)^+$
4.  $c^*(abcabc)$
5.  $(a^+b)^+(cd)^+$

Pour chacune des chaînes de caractères suivantes, déterminer quelles expressions régulières les acceptent.

- A. abcabc   B. abcd   C. abbccd   D. cabcab   E. aaccd   F. abaaabcd

**Exercice supplémentaire** Les expressions régulières non déterministes.

### Exercice 2.2

Compléter la DTD correspondant au premier exercice : appliquer des contraintes similaires à l'élément `courses` si possible.

### Exercice 2.3

Comparer les deux schémas, DTD et XML Schema. Quelles sont les principales différences ?