Structure logique

- La structure logique d'un document XML est une arborescence d'informations obtenue par un processeur XML à partir d'un contenu physique constitué d'une simple suite linéaire de caractères Unicode.
- Du point de vue logique, un document XML est composé :
  - D'éléments constituant la grammaire d'organisation ;
  - De textes constituant les données ;
  - éventuellement
    - De déclarations contenues dans un préambule,
    - De commentaires,
    - D'instructions de traitement.

25

Structure logique : préambule et déclarations (1)

- Le **préambule** (*prolog*) s'il est présent, précède le texte de balisage et les données textuelles d'un document. Il contient des déclarations qui peuvent être les suivantes :
  - La *déclaration XML*. Elle n'est pas obligatoire. La spécification XML précise cependant qu'elle devrait être placée au début d'un document XML. Si elle est présente, elle doit apparaître en premier ;

26

Structure logique : préambule et déclarations (2)

- La *déclaration de type de document* (Document Type Declaration : différente de la DTD). On ne la trouve que dans les documents XML traditionnels, et elle n'est obligatoire que pour la validation. Si elle est présente, elle doit se trouver dans le préambule et suivre la déclaration XML.
  La déclaration de type de document permet de référencer et/ou définir localement deux sortes d'entités :
  - Des définitions de type de document (DTD) contenant des déclarations de balisage ;
  - Des entités (autres que des DTD).

27

Structure logique : préambule et déclarations (3)

- La déclaration XML débute par « **<?xml** » et se termine par « **?>** ». Elle fournit trois informations sous la forme de pseudo-attributs.
  - version : cet attribut est obligatoire quand la déclaration XML est présente
    <?xml version="1.0" ?>
  - encoding : cet attribut indique le type de codage que doivent utiliser les processeurs XML afin de traduire des octets en caractères Unicode à la réception d'un document, ou de transformer des caractères Unicode en données binaires pour le transport. Il permet à des applications de se comprendre.
    <?xml version="1.0" encoding="ISO-8859-1" ?>

28

Structure logique : préambule et déclarations (3)

- standalone : cet attribut indique la nature autonome ou non d'un document. Un document est déclaré non-autonome lorsque l'entité document qui le représente est dépendante de déclarations de balisage externes
  <?xml version="1.0" encoding="**UTF-8**" standalone="no" ?>

- La déclaration de type de document est de la forme :

<!DOCTYPE document *MonTexte* "*document.dtd*" [*déclaration*]>

29

Structure logique : commentaires et instructions de traitement

- Les commentaires (Comment) peuvent être placés dans un fichier XML ou un fichier DTD :

  <!-- commentaire -->

- Instructions de traitement (Processing Instructions)
  - Instructions spéciales pour les applications qui manipulent les documents XML : <?cible arg1 arg2 …?>
  - cible : nom de l'application
  - arg1, arg2 : arguments passées à l'application
    <?xml-stylesheet type="text/css" href="style.css" ?>
    Cette instruction indique au navigateur d'afficher les données XML en appliquant la feuille de style : "style.css"

30

Structure physique

- Un document XML peut exister physiquement d'un seul bloc mais dépend le plus souvent de l'assemblage de plusieurs unités physiques existant séparément et appelées ***entités***.
- Les entités peuvent être schématisées comme des fichiers provenant de différents moyens de stockage situés à différents emplacements
- Chaque entité peut elle-même appeler d'autres entités

31

Structure physique

- En pratique, un document est représenté par une première entité (dite racine) qui peut contenir la totalité du document ou inclure le contenu des entités appelées afin de constituer un seul document logique.
- Les entités ont un *nom* qui permet de les appeler.
- Une entité racine (appelé aussi *entité document*) sert de point de départ pour le traitement d'un document.
- Les entités appelés dans le corps d'un document (et non dans une DTD) sont dénommées *entités générales*.

32

Structure physique : DTD

- Une DTD (Document Type Definition) est une sorte d'entité contenant une description formelle de la structure et du vocabulaire (noms de type d'éléments et noms d'attributs) d'un document.
- La spécification XML définit complètement l'écriture des DTD, mais une DTD n'est pas elle-même un document XML.

33

Structure physique : DTD

- Un processeur validateur lit la DTD avant de lire le document.
- Une DTD
  - Permet de définir le "vocabulaire" et la structure qui seront utilisés dans le document XML
  - Grammaire du langage dont les phrases sont des documents XML (instances)
  - Peut être mise dans un fichier et être appelée dans le document XML

34

Structure physique : DTD

- La DTD utilise la déclaration DOCTYPE qui référencie des déclarations soit internes ou externes
- **DTD interne** :
  - Les déclarations de la DTD sont placés dans le DOCTYPE après la déclaration XML
  - L'attribut standalone a la valeur "yes"

    <?xml version="1.0" standalone="yes">

    <!DOCTYPE document [déclarations]>
    <document > ... </document >

  - document : nom du type de document déclaré,
  - déclarations : déclarations d'éléments, d'attributs et d'entités…

35

Structure physique : DTD

- **DTD externe** :
- <!DOCTYPE document (SYSTEM "uri")>
  - Les déclarations sont placés dans un fichier séparé
  - Le nom du fichier est identique à la racine du document
  - L'attribut standalone a la valeur "no"
  - <!DOCTYPE document SYSTEM "http://ugb.sn/document.dtd">
- <!DOCTYPE document (PUBLIC "identifiant_public" "url"?)>
  - PUBLIC est utilisé lorsque la DTD est une norme ou qu'elle est enregistrée sous forme de norme ISO par l'auteur

36

Structure physique : DTD

- Déclaration de listes d'attributs
  - <!ATTLIST Nom_élément Nom_attribut type_attribut défaut>
  - <!ATTLIST document nd (1|2|3) "1">
- Type d'attributs :
  - CDATA : chaîne de caractères prise telle quelle
  - ID ou IDREF : renvoi dans le document
  - ENTITY ou ENTITIES : entité externe non XML
  - NMTOKEN : noms symboliques formés de caractères alphanumériques
  - NOTATION : attribut de notation

37

Structure physique : DTD

- Valeur par défaut :
  - la valeur en question
  - #REQUIRED : attribut obligatoirement présent et valeur spécifié par le créateur de l'instance
  - #IMPLIED : présence de l'attribut facultative
  - #FIXED "Valeur" : l'attribut prend toujours cette valeur

38

Structure physique : DTD

- Attributs d'un élément :
  - <!ATTLIST balise [attribut type #mode [valeur]]>
- Définit la liste des attributs pour une balise
  - <!ATTLIST enseignant
    nom CDATA #REQUIRED
    prenom CDATA #IMPLIED>
  - <!ATTLIST adresse
    ville CDATA #FIXED "Dakar">

39

Structure physique : DTD

- Déclaration d'un élément :
  <!ELEMENT balise valeur>
  - Décrit une balise **balise** qui fera partie du vocabulaire
  - ordre spécifié :
    <!ELEMENT balise (val1, val2, val3)>
  - ordre libre :
    <!ELEMENT balise (val1 | val2 | val3)>
  - Indicateur d'occurrence d'un élément
  - Contenu textuel déclaré comme PCDATA (parseable character data)

40

Structure physique : DTD

- Opérateurs d'occurrence
- Notations
  - (a, b) séquence
  - (a | b) liste de choix
  - a? élément optionnel [0,1]
  - a* élément répétitif [0,N]
  - a+ élément répétitif [1,N]
- Exemples
  - (nom, prenom, rue, ville)
  - (oui | non)
  - (nom, prenom?, rue, ville)
  - (produit*, client)
  - (produit*, vendeur+)

41

Structure physique : DTD

- Données dans un élément :
  - <!ELEMENT Donnees (#PCDATA)>
  - les données sont constituées par un flot de caractères
- Modèle mixte :
  - <!ELEMENT mixte (#PCDATA | val1 | val2)*>
  - dans cet exemple l'élément mixte est composé de données et d'éléments, le caractère de répétition est obligatoirement l'étoile
- Contenu libre :
  - <!ELEMENT libre ANY>
- Élément vide :
  - <!ELEMENT vide EMPTY>

42

Structure physique : DTD

- Les entités déclarées dans un document sont appelées "générales"
- Les entités déclarées dans une DTD sont appelées "paramètres" :
  - <!ENTITY % entite "valeur">
  - <!ENTITY % entite SYSTEM "URL">
  - <!ENTITY % entite PUBLIC "lien public">
- Référence à l'entité par &entite;
- Les entités paramètres déclarées dans la partie interne (entre les crochets "[" et "]" de DOCTYPE) ne peuvent pas être référencées dans une autre déclaration

43

Structure physique : DTD

- Liens hypertextuels internes : ID ou IDREF
  - Dans la DTD

    <!ELEMENT SECTION (#PCDATA|xref)*>

    <!ATTLIST SECTION TARGET ID #IMPLIED>

    <!ELEMENT xref EMPTY>

    <!ATTLIST xref ref IDREF #REQUIRED>
  - Dans l'instance

    <SECTION TARGET="cible"> contenu </SECTION>

    <SECTION> référence à la section

      <xref ref="cible"/>

    </SECTION>

44

Structure physique : DTD

- Liens hypertextuels externes vers des ressources cibles :
  - URL

    http://ugb.sn/cible.xml
  - pointeur XML :

    position absolue suivie d'une cascade de renvois relatifs qui spécifient la cible

    http://ugb.sn/cible.xml#root().child(1,titre)

    renvoi absolu

    #root().child(1,titre)

45

Structure physique : DTD

- Lien simple :

  <a xml:link="simple" href= "http://lien/s">...
  - inclus dans le document d'origine
  - une seule cible
  - unidirectionnel
  - on peut caractériser la cible (rôle, titre), l'origine (idem), la sémantique du lien (comportement lorsque le lien est traversé)…
- Lien étendu :
  - le but est de créer un lien multi-cible et de laisser l'application gérer la présentation des différentes cibles

46

Structure physique : DTD

CDATA	**Donnée composée de caractères**
Énumération	**Une série de valeurs dont on ne peut choisir qu'une**
ENTITY	**Une entité déclarée dans la DTD**
ENTITIES	**Entités multiples séparées par des espaces blancs et déclarées dans la DTD**
ID	**Un identificateur d'élément unique**
IDREF	**Valeur d'un attribut de type ID unique**
IDREFS	**IDREF multiples associés à des éléments NMTOKEN**
NMTOKEN	**Une marque associée à un nom XML**
NMTOKENS	**Des marques associées à des noms XML multiples séparées par des espaces blancs**
NOTATION	**Une notation déclarée dans la DTD**

47

Structure physique : Exemple DTD

```
<?xml version="1.0" encoding= "ISO-8859-1"
   standalone= "yes" ?>
<!DOCTYPE message [
    <!ELEMENT message (de, a, date, corps)>
    <!ELEMENT de (#PCDATA)>
    <!ELEMENT a (#PCDATA)>
    <!ELEMENT date (#PCDATA)>
    <!ELEMENT corps (#PCDATA|jour|heure|lieu)>
    <!ELEMENT jour (#PCDATA)>
    <!ELEMENT heure (#PCDATA)>
    <!ELEMENT lieu (#PCDATA)>
    <!ATTLIST message langue CDATA "Français">
]>
```

48

Structure physique : Entités analysables

- Les entités analysables ont un contenu textuel représentant du texte de balisage et des données textuelles.
- Une entité analysable peut elle-même être un document bien formé.
- Les appels d'entités sont traités par un processeur XML **validateur** pour générer un document global dans lequel le contenu des entités analysables se substitue aux appels de ces entités.
- Le document logique ainsi obtenu doit être bien formé pour être accepté.

49

Structure physique : Appels d'entités

- La notion de **référence** entre entités peut avoir deux sens différents :
  - On dit *référence* lorsqu'une entité annonce, dans sa déclaration de type de document, le nom d'une entité externe afin de pouvoir l'utiliser éventuellement ;
  - On dit plutôt *appel* lorsque, au cours de l'analyse d'un document par un processeur XML, une entité est appelé afin que son contenu soit développé dans le corps du document de l'entité appelante.
- L'appel d'une entité générale analysable de nom abcd s'écrit : « &abcd; ».

50

Structure physique : Appels d'entités

- Les entités (Entity) définissent une entité souvent utilisée (**entité interne**) :

  <!ENTITY nom_entite "valeur_entite">

- A chaque référence (Entity Reference) à une entité créée par l'auteur, de la forme &nom_entite;, dans le document, XML lui substitue valeur_entite

- Le mot SYSTEM indique que le contenu de l'entité est accessible par une URI (**entité externe**) :

  <!ENTITY cours SYSTEM "http://… /cours.xml">

51

Structure physique : Entités prédéfinies

- Un ensemble d'entités permet de masquer les caractères interdits en les remplaçant par des appels d'entités, afin de les utiliser dans la partie qui constitue les données du corps d'un document.
- Ces entités sont standardisées et sont reconnues par tous les processeurs XML sans qu'il soit nécessaire de les déclarer.
  - < (<) > (>) " (") ' (') & (&)

52

Structure physique : Appels de caractères

- Les appels de caractères désignent des caractères par leur valeur numérique Unicode décimale ou hexadécimale. Il est ainsi possible d'inclure dans un document XML des caractères non disponibles au clavier.
  - Par exemple, le symbole de copyright © s'écrit © ou ©.

53

Éléments et attributs

- Un élément a un nom, le plus souvent un contenu, et éventuellement des attributs.
- Le nom d'un élément est utilisé pour l'écriture de la balise de début et de la balise de fin qui délimitent le contenu de l'élément.
- Un élément peut contenir d'autres éléments (éléments fils), des données textuelles, les deux ou aucun des deux.

54

Éléments et attributs

- Des éléments ayant le même père sont dits frères (siblings).
- Selon la règle des documents bien formés, chaque élément n'a qu'un seul père.

55

Éléments et attributs

- Dans l'exemple déjà vu, l'élément mail a pour fils les éléments from, to, date, subject et body qui sont frères, mais n'ont pas eux-mêmes d'éléments fils.
- Excepté date, ils contiennent tous des données textuelles.
- Les éléments qui n'ont aucun contenu, comme l'élément date, sont des éléments vides.

56

Éléments et attributs

- On aurait pu décrire les mêmes données en les structurant différemment.
- Les éléments sont les briques servant à construire le corps des documents XML, puisque ce sont eux qui contiennent et organisent les données.
- La diversité des formes de structuration que procurent les différentes façons d'imbriquer les éléments et les données, ainsi que la rigueur imposée aux types d'organisation par la règle des documents bien formés, constituent la richesse de XML.

57

Éléments et attributs

- Autre représentation de l'exemple

```
<?xml version="1.0" encoding="UTF-8" ?>
<message to="toto@bidule.com"
   from="titi@truc.org">
   <date year="2000" month="10" day="06" />
   <subject>Bonnes nouvelles</subject>
   Salut les amis
</message>
```

- Utilisation d'attributs

58

Éléments et attributs

- Règles d'écriture des attributs :
  - Un élément peut avoir un nombre quelconque d'attributs, y compris zéro ;
  - L'ordre des attributs est indifférent ;
  - Les valeurs des attributs doivent être encadrées par les délimiteurs « " » ou « ' » ;
  - Les couples attribut/valeur sont séparés entre eux par des espaces, des tabulations ou des retours à la ligne ;
  - Les noms d'attributs doivent être uniques à l'intérieur d'un élément.

59

Éléments et attributs

- Élément racine
  - La règle des documents bien formés définit l'élément racine comme un élément qui contient tous les autres mais n'est contenu par aucun autre.
  - Il n'a pas d'élément père. On l'appelle élément document.
- Choix entre éléments et attributs
  - Il n'y a pas de réponse à cette question
  - Règle de base : lisibilité et compréhension réciproque

60