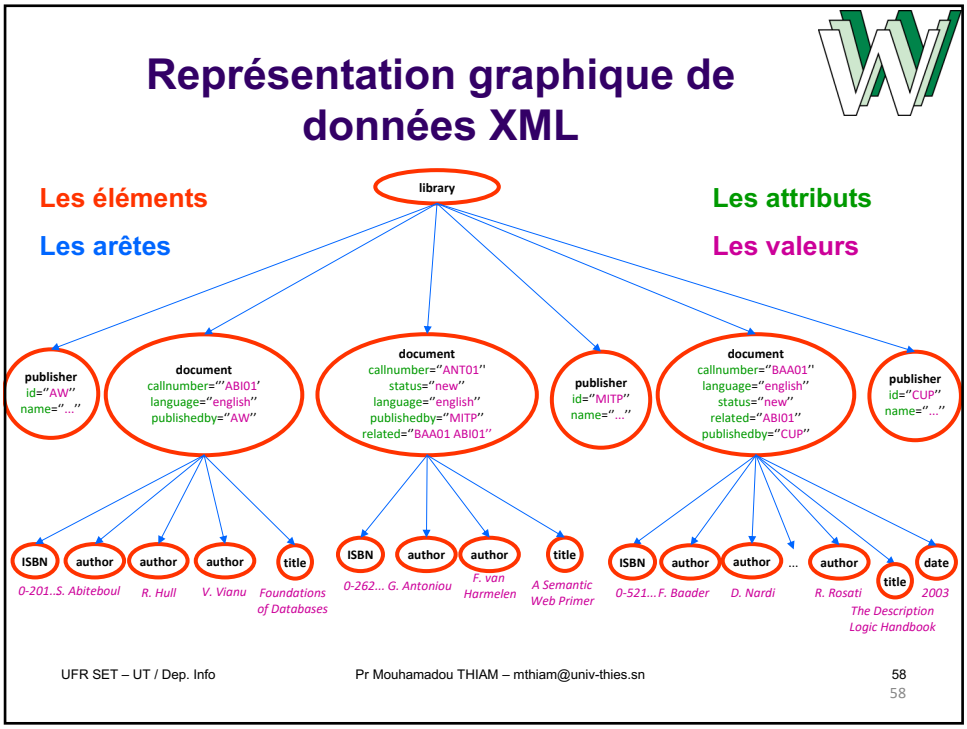
Introduction aux schémas de données XML

Schéma

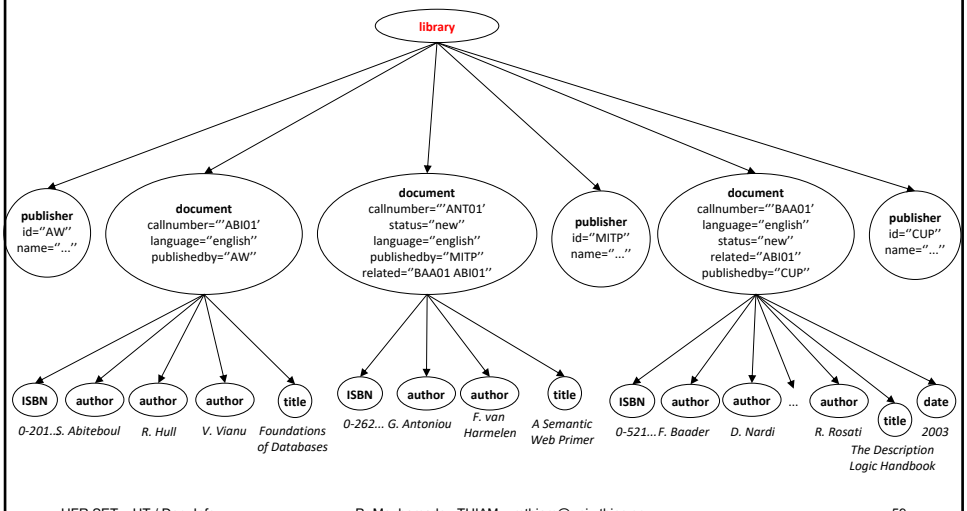
- Le schéma peut être commun à plusieurs documents
  - Facilite l'échange / le partage de données
- Le schéma permet une représentation plus fine des données
  - XML sans schéma : arbre
  - XML + schéma : graphe
- Dans le modèle de données XML, deux langages de définition de schémas
  - DTD
    - Pour des schémas simples
  - XML Schema
    - Pour des schémas complexes

Représentation graphique de données XML



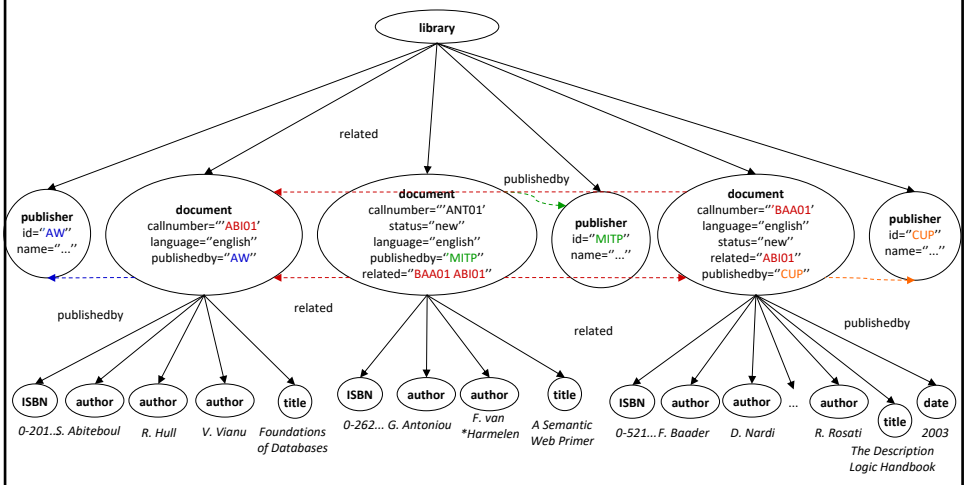
58

XML sans schéma : arbre



59

XML avec schéma : graphe



Schémas : validité

- Un schéma permet de vérifier la validité/cohérence des données.
- Le schéma peut être commun à plusieurs documents
  - Facilite l'échange / le partage de données
- Dans le cas général, on DOIT associer un schéma à tout document

DTD

Document Type Definition

62

DTD

- Les éléments qui décrivent un document peuvent être définis dans une **DTD** (Déclaration de Type de Document), mais ce n'est pas obligatoire.
- Un document XML est dit **valide** s'il est précédé de sa DTD et si sa description est conforme à cette DTD.
- Un document XML est dit **bien formé** s'il n'est pas précédé d'une DTD mais si sa description est syntaxiquement correcte.

63

DTD

- La déclaration d'un type de document (DTD) est composée d'une suite de déclarations :
  - déclaration d'éléments
  - déclaration des attributs d'un élément
  - déclaration d'entités

DTD d'un guide d'itinéraires à skis

- Un guide est composé d'un **titre**, d'une liste d'un ou plusieurs **auteurs**, d'un **éditeur**, d'une **année** et d'une liste d'un ou plusieurs **vallons**.
- Un titre, un auteur, un éditeur et une année sont des textes.
- Un vallon est composé d'un **nom**, d'une **introduction** et de la liste des **itinéraires** que l'on peut y réaliser (un ou plusieurs itinéraires).
- Un nom est un texte.

DTD d'un guide d'itinéraires à skis

- Une introduction est composée d'une liste d'un ou plusieurs **paragraphes**.
- Un paragraphe est un texte dans lequel sont insérés des **renvois** vers d'autres itinéraires et des **notes**.
- Une note est un texte donnant des **consignes** de prudence ou recommandant l'utilisation d'un **matériel** spécifique (crampons, piolet, etc.).

DTD d'un guide d'itinéraires à skis

- <!ELEMENT guide (titre, auteur+, editeur, année, vallon+)>
  - <!ELEMENT titre (#PCDATA)>
  - <!ELEMENT auteur (#PCDATA)>
  - <!ELEMENT editeur (#PCDATA)>
  - <!ELEMENT année (#PCDATA)>
  - <!ELEMENT vallon (nom, intro, itinéraire+)>
    - <!ATTLIST vallon id ID #REQUIRED>
    - <!ELEMENT nom (#PCDATA)>
    - <!ELEMENT intro (para+)>
      - <!ELEMENT para (#PCDATA | renvoi | note)*>
        - <!ELEMENT renvoi EMPTY>
        - <!ATTLIST renvoi cible IDREF #REQUIRED>
        - <!ELEMENT note (#PCDATA)>
        - <!ATTLIST note type (prudence | materiel) "prudence">
    - <!ELEMENT itinéraire (nom, alt, cotation, num, para+)>
      - <!ATTLIST itinéraire id ID #REQUIRED>
      - <!ELEMENT alt (#PCDATA)>
      - <!ELEMENT cotation (#PCDATA)>
      - <!ELEMENT num (#PCDATA)>

Déclaration d'un élément

- Un élément est défini par la déclaration :
  <!ELEMENT *nom modèle de contenu*>
- Un élément mixte pouvant contenir des éléments *T*1, …, *Tn* a pour modèle de contenu :

  (#PCDATA | *T*1 | … | *Tn*)*

Déclaration d'un élément

- Un élément composé d'une suite d'éléments *T*1, …, *Tn* a pour modèle de contenu l'expression régulière construite sur le vocabulaire {*T*1, …,*Tn*} à l'aide des opérateurs :
  - , (infixe) : concaténation
  - * + (suffixes) : 0 ou plusieurs répétitions et 1 ou plusieurs répétitions
  - ? (suffixe) : optionalité
- Un élément vide a pour modèle de contenu EMPTY.
- Un élément de contenu quelconque a pour modèle de contenu ANY.

Déclaration des attributs d'un élément

- A chaque type d'élément est attaché un ensemble d'attributs.
- Une définition d'attributs a la forme suivante :

  <!ATTLIST *nom-élément nom-attribut1 type1 déclaration-de-défaut1*
  …
  nom-attribut2 type2 déclaration-de-défaut2>
- où :
  - le type est celui des valeurs de l'attribut,
  - la déclaration de défaut spécifie si la valeur de l'attribut doit être ou non présente dans le document et fournit éventuellement une valeur par défaut.
- des éléments de type différent peuvent avoir des attributs de même nom.

Déclaration des attributs d'un élément

- La déclaration de défaut peut être :
  - #REQUIRED : l'attribut doit être présent dans la balise de l'élément,
  - #IMPLIED : l'attribut est facultatif, *valeur* : valeur à affecter à l'attribut s'il est absent de la balise de l'élément (valeur par défaut),
  - #FIXED *valeur* : valeur que doit avoir l'attribut s'il est présent dans la balise de l'élément ou qui lui sera affectée s'il est absent de cette balise.
- Les déclarations de défaut sont prises en compte par un analyseur XML afin de compléter le document analysé.

Déclaration des attributs d'un élément

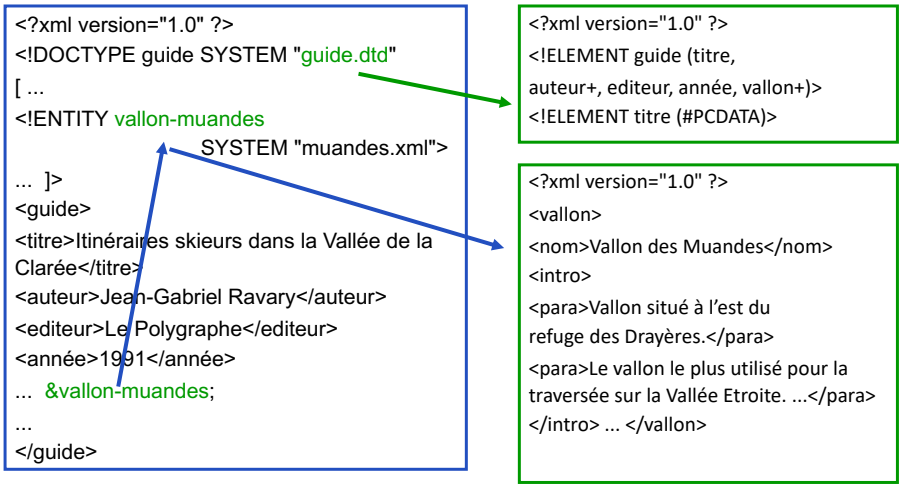
- Le type de valeur peut être :
  - CDATA : texte,
  - ID : nom identifiant l'élément dans le document (que nous appellerons **identificateur**),
  - IDREF ou IDREFS : identificateur ou suite d'identificateurs (séparés par une suite de séparateurs : espace, tabulation),
  - NMTOKEN ou NMTOKENS : nom ou suite de tokens de nom,
  - (*nom*1 | … | *nomn*) : un des tokens de nom énumérés.
  - ENTITY, ENTITIES et NOTATION

Réalisation physique d'un document : les entités

- Un document XML est physiquement découpé en **entités**.
- Une entité est un fragment nommé de document.

- On distingue :
  - les **entités générales** qui sont des fragments nommés de l'élément du document,
  - les **entités paramètres** qui sont des fragments nommés de DTD,
  - les **entités prédéfinies** qui sont des caractères réservés de XML,
  - les **entités caractères** qui sont des caractères du jeu de caractères universel UNICODE, nommés par leur code numérique.

Réalisation physique d'un document : les entités



Remplacement des entités

- Une référence à une entité est remplacée par sa valeur lorsque l'élément ou la DTD qui la contient est traité par un parseur XML.
- Ce remplacement pourra entraîner des remplacements en cascade si cette entité contient elle-même des références à des entités et ainsi de suite.

Entité générales

- **Déclaration** d'une entité générale :
  - **interne** (enregistrée dans sa déclaration)
    <!ENTITY *nom* "*entité*">
  - **externe** (enregistrée dans un fichier externe à celui de sa déclaration)
    <!ENTITY *nom* SYSTEM "*nom du fichier contenant l'entité*">
  - Par exemple :

<!ENTITY ref "refuge">

<!ENTITY vallon-muandes SYSTEM "mon_site/muandes.xml">

Entité générales : Référence

- **Référence** à une entité générale : &*nom de l'entité*;
  - Par exemple :

<para>S'élever au-dessus du &ref; des Drayères …</para>

  - est équivalent à :

<para>S'élever au-dessus du refuge des Drayères …</para>

Entité paramètres

- **Déclaration** d'une entité paramètre :
  - interne (enregistrée dans sa déclaration)
    <!ENTITY % *nom* "*entité*">
  - externe (enregistrée dans un fichier externe à celui de sa déclaration)
    <!ENTITY % *nom* SYSTEM *nom du fichier contenant l'entité*>
  - Par exemple :
    <!ENTITY % identificateur ID #REQUIRED>

Entité paramètres : référence

- **Référence** à une entité paramètre :
  %nom;
  - Par exemple :
  <!ATTLIST renvoi cible %identificateur;>
  - au lieu de :
  <!ATTLIST renvoi cible ID #REQUIRED>

Entité prédéfinies

- Les caractères < > & ' " qui sont des délimiteurs XML peuvent être remplacés dans un texte par une référence à une entité prédéfinie.
- Ces entités sont les suivantes :
  < référence au caractère <
  > référence au caractère >
  & référence au caractère &
  ' référence au caractère '
  " référence au caractère "
- Par exemple, la phrase :
  « L'expression <ALT>2794</ALT> est un élément XML. »
- peut être représentée par l'élément suivant :
  <phrase>L'expression <ALT>2794</ALT> est un élément XML.</phrase>

Entités caractères

- Un caractère non disponible sur la station de travail peut être représenté par son code **Unicode** en décimal ou en hexadécimal, sous la forme d'une référence a une entité :
  - &#*code décimal*;
  - &#x*code hexadécimal*;
- Par exemple :
  - & référence au caractère &
  - Φ référence à la lettre grecque Φ

Organisation d'un document XML valide

- Un **document XML valide** est composé d'une **entité document** (sans nom) et d'un ensemble d'**entités externes**.
- L'entité document est composé d'un **prologue** et de l'élément du document.
- Le prologue est composé d'une **déclaration XML** et d'une DTD.
- **La déclaration XML indique :** la version de XML, le jeu de caractères et l'éclatement ou non du document en plusieurs entités externes.
- La DTD est constituée d'une **partie interne** placée dans l'entité document et d'une **partie externe**, enregistrée dans un fichier à part dont le nom est déclaré dans l'entité document.

Organisation d'un document XML valide

- La partie interne de la DTD, l'élément du document et les entités externes peuvent appeler des entités externes. Ces appels doivent être non récursifs et non circulaires.
- L'organisation d'un document bien formé est similaire à celle d'un document à l'exception de la DTD qui est :
  - soit absente,
  - soit présente mais ne contient que des déclarations d'entités générales.

Document XML monofichier

- <?xml version="1.0" encoding="…"
  standalone="yes" ?>
- <!DOCTYPE *nom* [
  déclarations
  ]>
  <*nom*>
  …
  <*nom*>
- où :
  - L'attribut standalone='yes' indique que le document est contenu en entier dans le fichier.
  - Le nom de l'élément du document doit être identique à celui de la DTD

Document XML multifichiers

- <?xml version="1.0" encoding="…"
  standalone= "no" ?>
- <DOCTYPE *nom* SYSTEM *nom_fichier* [*partie interne de la DTD*]>
- *élément du document*
- où :
  - *nom_fichier* est le nom du fichier contenant la partie externe de la DTD ;
  - l'attribut standalone="no" indique qu'il est fait appel à des entités externes soit dans la partie interne de la DTD, soit dans l'élément du document ;
  - le nom de l'élément du document doit être celui de la DTD ;
  - une entité externe peut débuter (et c'est conseillé) par une déclaration XML sans attribut standalone.

END

?

86

86

Codage des caractères

87

87

Structure globale de document XML

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nom SYSTEM "fichier.dtd ou URL" [
déclarations
]>

... corps du document ...
```


UCS : un jeu de caratères universel

- Le texte est stocké dans la mémoire de l'ordinateur sous forme de bits groupés en octets.
- La norme ISO 10646 en accord avec **l'Unicode** définit un jeu de caractères universel : l'UCS (« Universal Character Set ») qui permet de représenter les caractères de toutes les langues actuelles mais aussi anciennes.
- Chaque caractère UCS est identifié par un code qui est un nombre représenté sur 4 octets (2^{32} - 1 positions).
- Les 65 536 premières positions de l'UCS (c-à-d. les deux octets de poids faible) forment le BMP (« Basic Multilingual Plane ») et codent les jeux de caractères les plus courants (latin, grec, arabe, etc.). D'où deux codages :
  - UCS-4 : totalité de l'UCS,
  - UCS-2 : BMP.

Codages de transformation

- Plusieurs codages de transformation ont été définis :
  - **UTF-8 :** permet de coder les caractères de l'UCS en longueur variable (de 1 à 4 octets) en codant sur un octet les caractères ASCII qui sont les plus fréquents.

UCS-32BE	UTF-8			
	Octet 1	Octet 2	Octet 3	Octet 4
00000000 00000000 0xxxxxxx 00000000 00000yyy yyxxxxxx 00000000 zzzzyyyy yyxxxxxx 000uuuzz zzzzyyyy yyxxxxxx	0xxxxxxx 110yyyyy 1110zzzz 11110uuu	10xxxxxx 10yyyyyy 10zzzzzz	10xxxxxx 10yyyyyy	10xxxxxx

  - **UTF-16 :** permet d'inclure des caractères de l'UCS-4 dans une chaîne codée en UCS-2.

Déclaration de codage

- Toutes les applications XML doivent accepter les codages UTF-8 et UTF-16.
- D'autres codages peuvent être acceptés, tels que le codage ISO-8859-1 (« ISO-Latin »).
- Le codage des caractères d'une entité doit être déclaré dans la déclaration XML de cette entité, comme valeur de l'attribut **«encoding»**. S'il ne l'est pas, l'application considérera par défaut être en présence d'un codage UTF-8.