XML

C. Delobel, V. Benzaken, E. Waller

Historiques

- IBM : 1969, GML Generalized Markup Language
- SGML : 1978, norme Iso en 1986
- ASN/1 : normalisation des données biologiques
- XML

Objectifs

- Définir des structures logiques pour décrire des documents.
- Séparer la structure logique de la présentation du document sur un support externe (Web, papier).
- Permettre des recherches d'information.
- Garantir l'indépendance vis à vis des matériels et des logiciels.

HTML versus XML

```
<h1> Personne </h1>
<p> <b> pierre </b> 30 ans <i>pierre@lri.fr </i> </p>
<p> <b> alice </b> 22 ans <i> alice@inria.fr </i> </p>

<employe>
        <description> Personne </description>
                <listepersonne>
                        <personne>
                                <nom> pierre </nom>
                                <age> 30 </age>
                                <email> pierre@lri.fr </email>
                        </personne>
.............................................................
```

Principes élémentaires

- L'unité est le document composé de caractères délimité par des marques (balises).
- La liste des marques peut être spécifiée en dehors du document dans une DTD.
- Un document est vu comme une arborescence d'éléments où les nœuds correspondent à un début et une fin de marques.
- Les éléments peuvent posséder des attributs valués.
- Il est possible d'avoir des liens entre document.

Attributs XML

- Les attributs sont des propriétés attachées à une marque.
- La liste des attributs pour un élément est fixe.
- Le document XML contient des paires nom d'attribut et valeur.
- Attention : le concept d'attribut XML est différent de celui du modèle relationnel car la même information peut être représentée de façon différente.

Exemple de modélisation

```
<personne> <nom> pierre </nom>
        <age> 30 </age>
        <email> pierre@lri.fr </email>
</personne>
```

```
<personne nom = "pierre" age= "30" email = "pierre@lri.fr" />
```

```
<personne age = "30">
        <nom> pierre </nom>
        <email> pierre@lri.fr </email>
</personne>
```

Donnée semi-structurée

<ssd-exp> ::= <val> | oid <val> | oid
<val> ::= atome | <complexval>
<complexval> ::= { label : <ssd-exp>,
label : <ssd-exp>,
....
label : <ssd-exp>
}

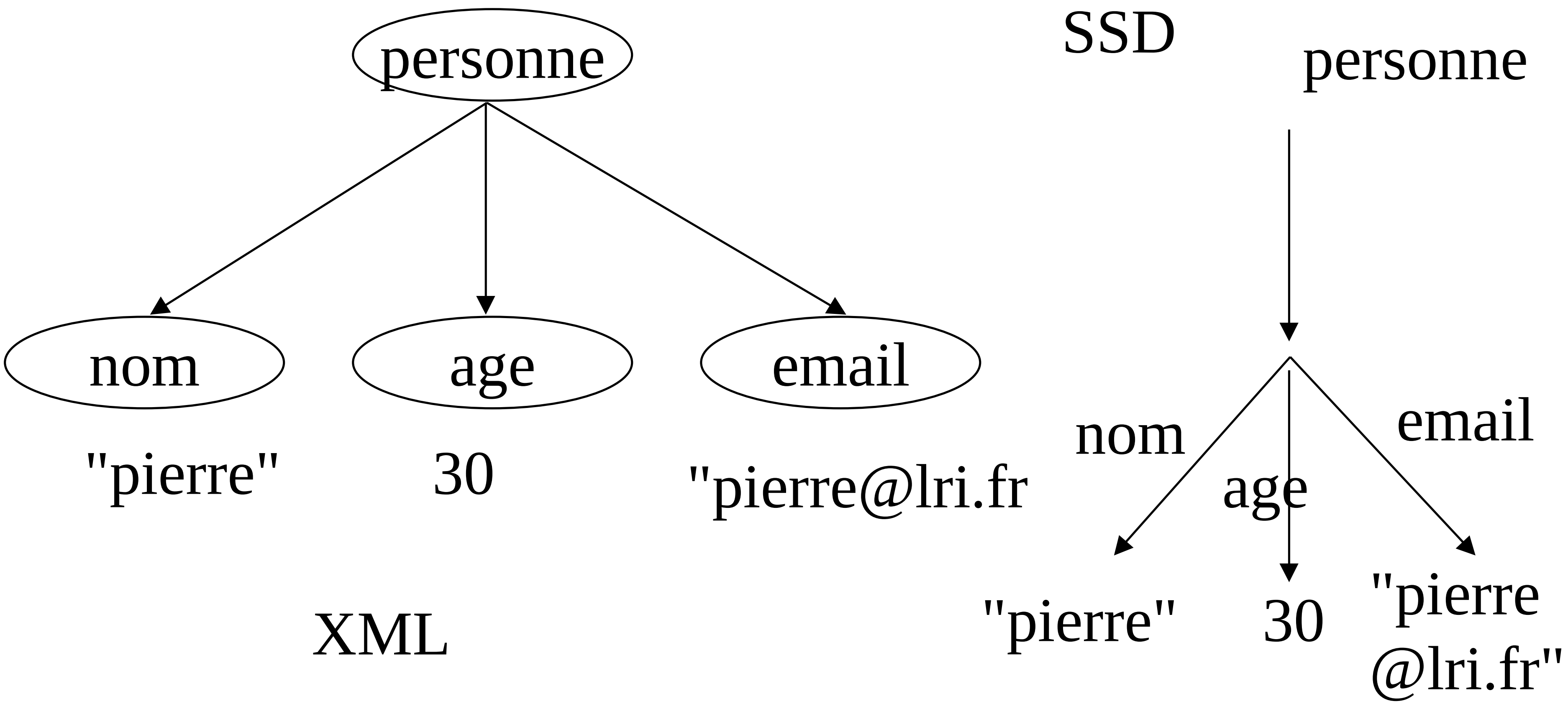
une expression ssd est valide : si un oid est utilisé alors il est défini.

XML et ssd-exp

```
<personne> <nom> pierre </nom>
           <age> 30 </age>
           <email> pierre@lri.fr </email>
</personne>
```

```
{personne : {nom : "pierre",
             age : 30,
             email : "pierre@lri.fr"
            }
}
```

Représentation graphique



Référence XML

<personne> <nom> pierre </nom>
<age> 30 </age>
<email> pierre@lri.fr </email>
<villede idref = &v1 />
</personne>

<ville id=&v1>
<nomdeville> paris </nomdeville>
<pays> france </pays>
</ville>

Références cycliques

```
<personne id = &1>
        <nom> adam </nom>
        <epoux idref= &2 :>
</personne>

<personne id = &2>
        <nom> eve  </nom>
        <epoux idref= &1 :>
</personne>
```

DTD : est-ce un schéma ?

- DTD pour Document Type Definition
- La DTD définit le type d'un document.
- Le langage pour décrire les DTD est proche des BNF.
- La DTD définit les éléments composant un document, ces éléments peuvent être optionnels, répétés, ou correspondre à différents cas, les éléments peuvent posséder des attributs et référencés d'autres documents.

Exemple de DTD

```
<!DOCTYPE db [
        <!ELEMENT db (personne*)>
        <!ELEMENT personne ( nom, age, email)>
        <!ELEMENT nom (#PCDATA)>
        <!ELEMENT age (#PCDATA)>
        <!ELEMENT email (#PCDATA)>
]>
```

0,1, ..., n :e * ;   1,..., n : e+ ; 0, 1 : e? ; e | e' ;

Exemple (suite)

```
<!DOCTYPE famille [
        <!ELEMENT famille (personne)*>
        <!ELEMENT personne (nom)>
        <!ELEMENT nom (#PCDATA)>
        <!ATTLIST person id ID #REQUIRED
                         père IDREF #IMPLIED
                         mère IDREF IMPLIED
                         enfants IDREFS IMPLIED>
]>
```

Les liens XML

- Le concept n'est pas encore totalement stabilisé. Il est voisin de celui de html.
- Xlink référence vers un autre document externe.
- Xpointeur référence vers une portion d'un document en utilisant des chemins.

Langages de requêtes

- Une DTD n'est pas un schéma de base de données, il faut donc introduire des concepts permettant une plus grande souplesse.
- Les propositions :
  - XML-QL
  - YATL
  - Lorel
  - XQL

C. Delobel, V. Benzaken, E. Waller

Les principes

- Les requêtes doivent comportées trois parties :
  - un motif de filtrage
  - une condition de filtrage
  - une fonction de construction du résultat
- Les fonctionnalités :
  - fonction de restructuration
  - fonction de composition des documents à partir de plusieurs type de documents.
  - variables de marque ou de chemin pour permettre l'interrogation sans une connaissance précise de la structure.

Exemple de DTD

```
<!ELEMENT bib (book*) >
<!ELEMENT book (title, (author+ |editor+), publisher, price)>
<!ATTLIST book year #CDATA>
<!ELEMENT author ( last, first)>
<!ELEMENT editor ( last, first, affiliation)>
<!ELEMENT title #PCDATA>
<!ELEMENT last #PCDATA>
<!ELEMENT affiliation #PCDATA>
<!ELEMENT publisher #PCDATA>
<!ELEMENT price #PCDATA>
```

Exemple de requêtes XML-QL

```
construct <bib> {
 where
    <bib>
        <book year = $y>
             <title> $t </title>
             <publisher> <name> Addison <name> </publisher>
        </book>
   </bib> in "www.bn.com/bib.xml
   $y > 1991
 construct <book year = $y> <title> $t </title> </book>
} </bib>
```

Exemple YATL

```
make
        bib [ *book [@year [$y],
                     title [$t]]]
match "ww.bn.com/bib.xml with
        bib [*book [ @year [$y],
                     title [$t] ],
                     publisher [name [$n]]]
where $n = "addison-wesley" and $y > 1991
```

Bibliographie

- Serge Abiteboul et al, Data on the Web, Morgan Kaufmann, 1999.
- Alain Michard, XML, Eyrolles, 1999.
- A. Deutsh et al, A query language for XML, International Web Conference, 1999.
- S. Cluet, C. Delobel, J. Siméon, Yours mediators need data conversion, ACM SIGMOD, 1998.
- S. Abiteboul et al, The lorel query language, International Journal on Digital Libraries, 1997.
- J. Robie, XQL 99 proposal, 1999, http://metalab.unc.edu/xql/xql-proposal.html