XML:
eXtensible Markup Language

Projet ACACIA

Olivier.Corby@inria.fr

INRIA

Plan

- Introduction
- DTD XML (Document Type Definition)
- Outils
- Formalismes associés:
  - XLL: Liens hypertextes
  - XSL: Feuilles de styles
  - RDF: Meta informations
- XML et Mémoire d'entreprise

INRIA

Introduction

- Version simplifiée de **SGML**, *SGML Light*, conçu pour le Web
- SGML: Standard Generalized Markup Language, 1986
  - Issu lui même de travaux d'IBM: Generalized Markup Language
  - **HTML**, élaboré selon la norme SGML: 1992

Simplifications:

- La DTD (Document Type Definition) n'est pas obligatoire
- Les options sont supprimées

INRIA

XML et W3C

- XML élaboré par la communauté SGML: été 96
- Spécifié au sein du Consortium W3C
- Recommandation du W3C, février 1998
  - `http://www.w3.org/XML`

INRIA

XML: un métalangage

- Générateur de langages de description de documents structurés.
- Pour publier des **documents** structurés sur l'Internet
- Pour échanger des **données** structurées
- Séparer:
  - données structurées
  - présentation

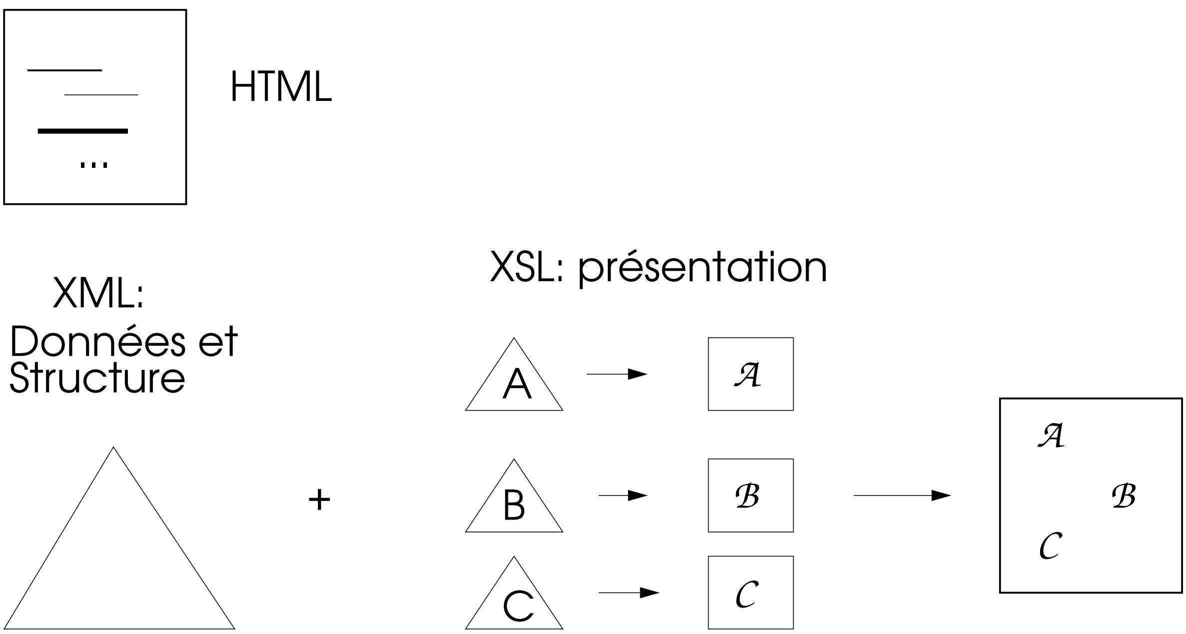
Exemple:

```
<bibliography>
    <book id="LCCN=36027176">
        <author>John Maynard Keynes</author>
        <title>The General Theory of Employment,
                Interest and Money
        </title>
        <publisher>Macmillan</publisher>
        <year>1936</year>
    </book>
</bibliography>
```

XML vs HTML

- Définir ses propres balises
- Pas de limitation d'imbrication des entités XML
- Validation possible par rapport à une DTD
- Uniquement la structure du document, pas de format de présentation
  - XSL: XML style sheets
- Amélioration du système de liens hypertextes

Structure vs Présentation



Avantages

- **Réutiliser** des informations
- Publier indépendamment des medias
- Imprimé et numérique à partir de la même source:
  - *write once, publish everywhere*
- Indépendant de formats propriétaires
- Indépendance par rapport aux vendeurs et aux plates-formes
- XML + XSL peut remplacer formats de traitement de texte et logiciel de publication
- **Standard** pour représenter les données: on peut changer d'outils

Avantages (suite)

- Améliorer la **recherche d'information** sur le Web:
  - recherche sur des balises et des DTD spécifiques
  - recherche sur le Web comme dans une base de données
- Une communauté peut définir un **standard d'échange** d'information:
  - se mettre d'accord sur une DTD
  - échanger des données XML, sans la DTD

Types d'Application

- Client Web médiateur de plusieurs **sources hétérogènes**
- Décharger le serveur, **charger le client** d'une partie du traitement:
  - **présentation** des données
  - accès aux données pour faire des calculs, des transformations, etc.
- Présenter **différentes vues** des mêmes données, sans les recharger à chaque fois
- **Adapter** le format de présentation à l'utilisateur

La galaxie XML

- **XML, XML name spaces**
- **XLL**, XML Linking Language: liens hypertextes
  - **XPointer**: langage de chemins pour naviguer dans les documents
- **DOM**: Document Object Model
  - API d'accès aux documents
- **XSL**, XML Style sheets Language: feuilles de style
  - **CSS,** Cascading Style Sheets

Formalismes XML

- **RDF:** Resource Description Framework, méta informations pour annoter des documents
  - **- RDF Schema**
- **MathML** (W3C)
- **SMIL**: Synchronized Multimedia Integration Language (W3C)
- **XMI**: Metadata Interchange Format (OMG)
- **XML/EDI**: Electronic Data Interchange, Commerce électronique basé sur UN/EDIFACT
- **TIM**: Telecom Interchange Markup (ATIS)
- **CML**: Chemical Markup Language
- **ThML**: Theological Markup Language

INRIA

Exemple

```
<book>
    <title>A Theory of Justice</title>
    <author>John Rawls</author>
    <abstract>A Theory of Justice as fairness, ...
    </abstract>
    <chapter>
       <title>Introduction</title>
       <section> ...
       ...
    </chapter>
    ...
</book>
```

INRIA

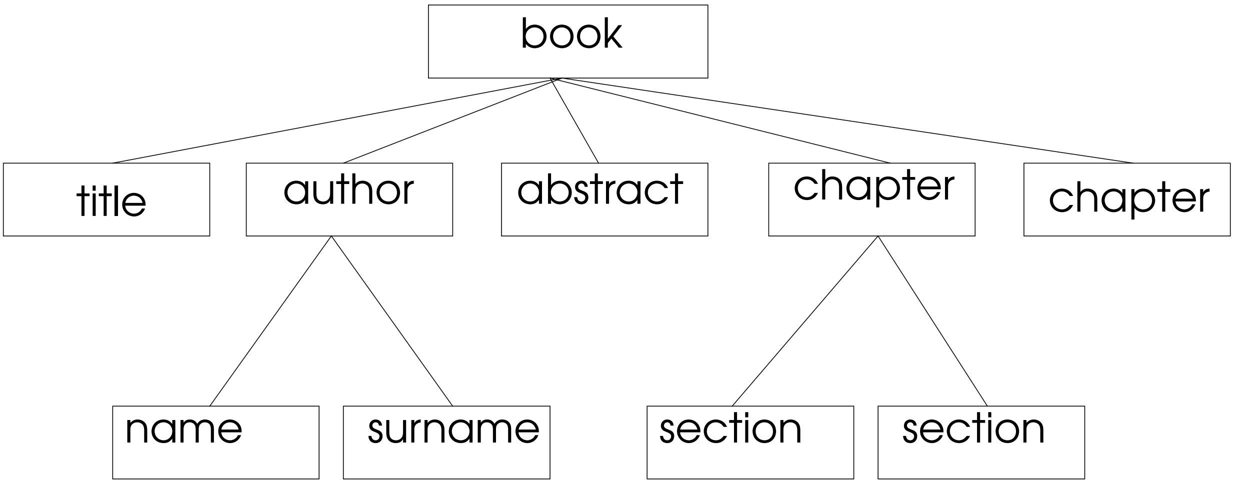
Document XML

- Document **bien formé**: well formed. (syntaxe)
  - balises correctement emboîtées, pas de '<' dans le texte, etc.
- Document **valide**: (sémantique) vérifie des contraintes supplémentaires
  - avoir une DTD et s'y conformer
- Document composé de:
  - **Texte**: Parsed Character data (PCDATA)
  - **Balises**: Markup

DTD: Document Type Definition

- DTD: description formelle de la **structure** du document
- Locale, importée ou les deux
- DTD contient:
  - element type
  - attribute list
  - entity
  - notation
- DTD est facultative
- Un document qui déclare une DTD doit s'y **conformer**

DTD: Structure



DTD

```
<?xml version="1.0"?>

<!DOCTYPE Library
      SYSTEM "http://somewhere.net/book.dtd">

<!DOCTYPE Library [ ... ]>

<!ELEMENT book (author, title, publisher, text)>

<!ELEMENT author    (#PCDATA)>
<!ELEMENT title     (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
```

Element

```
<!ELEMENT text ((para|figure)+)>

<!ELEMENT nihil EMPTY>
   <nihil/>

<!ELEMENT cool ANY>
```


Attribute list

```
<!ATTLIST book id ID #REQUIRED>

       <book id="V0-98/04">...</book>


<!ELEMENT reference(#PCDATA)>
<!ATTLIST reference value IDREF #IMPLIED>

       <reference value="V0-98/04">
          Un livre passionnant ...
       </reference>
```

Attribute list (2)

```
<!ATTLIST title
        form (bold|italic|normal) "normal" #IMPLIED>

         <title form="italic">
             A Theory of Justice
         </title>


<!ATTLIST author name CDATA #REQUIRED>

        <author name="Jean-Sol Partre"/>
```


Exemple:

```
<!ELEMENT form((select|input)+)>

<!ATTLIST form
    method (get|post) #REQUIRED
    action %URL; #REQUIRED
    target CDATA #IMPLIED>


<form method="get" action="/cgi/pouet.pl">
```

Entity

- Entité définie, à laquelle est associé un nom
  - parsed entity (XML)
  - unparsed entity (image, etc.) : référencée par son nom dans un attribut de type ENTITY
- General parsed entity:

```
<!ENTITY cpr "Copyright INRIA 1998">

<text>du texte libre ... &cpr; </text>

<text>du texte libre ... Copyright INRIA 1998 </text>
```


Entity

- Parameter parsed entity: utilisée dans la DTD pour factoriser/ customiser des portions de documents

```
<!ENTITY % attr-list
    "name CDATA #IMPLIED id IDREF #REQUIRED">

<!ELEMENT Test (#PCDATA)>

<!ATTLIST Test %attr-list; >

<!ATTLIST Test
     name CDATA #IMPLIED id IDREF #REQUIRED>
```

Unparsed entity

- Pour intégrer des images, etc:

```
<!ENTITY icon SYSTEM "../icons/test.gif" NDATA gif>

<!ELEMENT figure EMPTY>
<!ATTLIST figure image ENTITY #REQUIRED>

<figure image="icon"/>
```


Notation

- Notation: pour déclarer un processeur pour des éléments non XML: image, audio, etc.

```
<!NOTATION gif SYSTEM "gifprocessor">
```

Autres

- Commentaires:

```
<!-- comment -->
```

- CDATA: pour considérer comme du texte une portion de document contenant des balises:

```
<![CDATA [
       ici du texte avec des balises est reconnu comme
       du texte, exemple: ici <a> n'est pas une balise
]]>
```


Outils

Browser XML

- **Documents** structurés:
  - définir des feuilles de styles et le navigateur présente le document: CSS, XSL
- **Objets** structurés:
  - Ecrire des applets, par exemple en Java, qui font les traitements appropriés.

Outils

- **Parser** XML comme ***XML for Java*** d'IBM, **XP** de James Clark
  - Analyse un document XML et construit un arbre
- Document accessible via une API:
  - **DOM**, Document Object Model (W3C),
  - **SAX**: Simple API for XML
- XSL:
  - XT de James Clark
  - Koala XSL Engine

Outils (2)

- Jumbo: browser "graphique"
- Internet Explorer, msxml, XML notepad
- Mozilla (Netscape) fin 1998 ?
- Editeurs XML: FrameMaker 5.5, Grif, Adept Editor de ArborText, ..
- DataBase: ObjectStore

XLL: XML Linking Language

- Working Draft de Mars 98
- **Etendre** les liens hypertexte de **HTML**
- Inspiré de HyTime et TEI
- Associer de la **sémantique** aux liens et aux ressources
- Liens posés de l'extérieur du document
- Liens bidirectionnels
- Liens n-aires

XML Link

- Réalisé par un attribut réservé:
  - `xml:link`
- Tout élément XML peut "être" un lien hypertexte
- Exemple:

```
<reference xml:link="simple"
    href="http://there.somewhere.net">
    Un peu de texte ...
</reference>
```

XPointer

- Langage pour désigner des chemins dans des documents XML
- Peut être utilisé dans une URL pour désigner un élément de document sans insérer d'ancre
- Exemple: 3ème fils de la 4ème section du 2ème chapitre:

```
CHILD(2,CHAPTER).(4,SECTION).(3)
```

- Peut désigner une zone avec un début et une fin:

```
span(First,Last)
```


CSS

- Cascading Style Sheets, standard W3C pour HTML
- S'applique à XML pour un formattage "élémentaire"

```
<STYLE type="text/css">
    header, chapter, section {display:block}
    author {font-weight: bold}
</STYLE>
```

XSL

- XML Style Sheets, basé sur DSSSL (formatage de SGML)
- Spécifier la présentation d'un document XML

- **Transformation** de documents
- **Mise en page** de documents
  - Inclut HTML comme langage de mise en page
- XSL permet d'accéder à toutes les parties de la structure du document
- Mise en page pour affichage numérique et support papier

INRIA

Exemple

```
<chapter>
<title> ... </title>
<body>
...
</body>
</chapter>
```

```
<xsl:template match="chapter">
    <h1><xsl:value-of expr="title"/></h1>
    <xsl:process select="body"/>
</xsl:template>
```

INRIA

DOM: Document Object Model

- Standard W3C
- **Interface** permettant d'accéder à la **structure** et au **contenu** d'un document XML
- Définit une API en IDL, Java et ECMAScript
- Programmer une application avec l'API
  - Choisir un parser ou un navigateur XML qui supporte l'API DOM

INRIA

RDF: Resource Description Framework

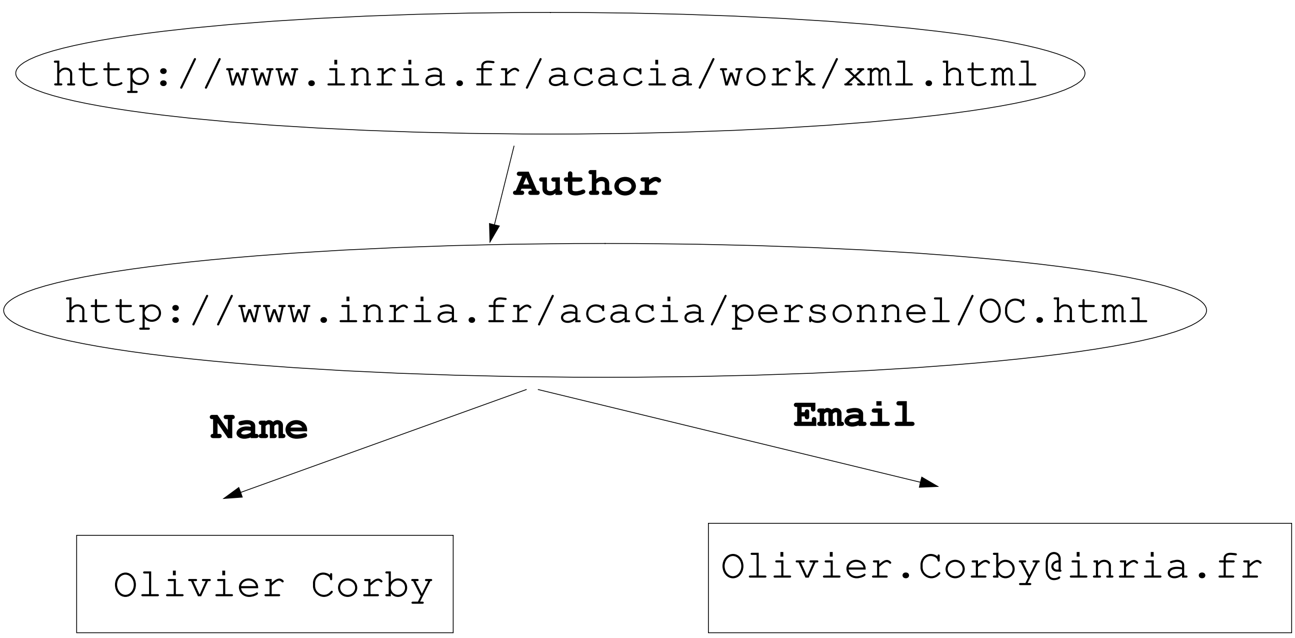
- Pour exprimer des méta informations sur des ressources de l'Internet (pages Web, etc.)
- Aide pour de meilleurs moteurs de recherche de ressources
- Catalogage pour décrire le contenu
- Aider des agents intelligents pour faciliter le partage et l'échange de connaissances

INRIA

RDF (suite)

- RDF:
  - modèle pour exprimer des metadata
  - syntaxe en XML
- RDF est indépendant d'un domaine
- Il peut fournir un ensemble de classes dédiées à un domaine particulier: RDF **Schema**
- RDF: *a simple frame system*
- Un mécanisme de raisonnement pourrait être implémenté au dessus de RDF (cf WD)

Exemple



Exemple

```
<rdf:Description
   about="http://www.inria.fr/acacia/work/xml.html">
   <s:author>
     <s:Person about="http://www.inria.fr/
         acacia/personnel/OC.html">
         <name>Olivier Corby</name>
         <email>Olivier.Corby@inria.fr</email>
     </s:Person>
   </s:author>
</rdf:Description>
```


XQL: XML Query Language

- Pour interroger des serveurs Web/XML
- Interrogation et transformation

```
WHERE
<book>
  <publisher><name>Addison-Wesley</name></publisher>
  <title> $t </title>
  <author> $a </author>
</book> IN "www.a.b.c/bib.xml"
CONSTRUCT
<result>
  <author>$a</author>
  <title>$t</title>
</result>
```

X Files: Sites de référence

- W3C: World Wide Web Consortium

  `http://www.w3.org/XML`

- OASIS: Organization for the Advancement of Structured Information Standards

  `http://www.oasis-open.org/cover`

XML et Mémoire d'entreprise

- Synthèse entre **SGML et le Web**,
  - Diffusion de documents structurés sur Intranet/Internet
- Synthèse entre **documents et données**
  - Documents composites
  - Intégrer documents et données
- Bases de données et base documentaire délivrant du XML
- Standard d'échange et d'intégration de données hétérogènes
- On peut enrichir une base avec des données XML

XML et ME (2)

- Write once, publish everywhere:
  - même source d'information pour diffusion numérique et papier
- **Adapter la présentation** en fonction de l'utilisateur, du contexte, du media, etc.
- Plusieurs vues sur le même document
- Format ouvert et simple: intéressant sur le long terme

XML et ME (3)

- **Recherche d'information** guidée par des balises (vs recherche plein texte)
  - Facilite le travail d'agents intelligents de recherche d'information
- L'entreprise peut définir un **vocabulaire interne** pour les balises XML
- **Documents intelligents**: documents + ontologies + capacités de recherche et d'inférence
- **Liens hypertextes** sophistiqués
- Documents actifs: documents + applet Java