

Pour commencer

- SGML
 - Standard Generalized Markup Language
 - Norme générique issue du document technique
- HTML
 - Hypertext Markup Language (tag = balise)
 - Application de SGML au Web
- Xml
 - eXtensible Markup Language (tags HTML + vos tags)
 - Une nouvelle version de SGML pour le Web et...
 - élargir la nature des documents à échanger sur le Web
 - faciliter l'interopérabilité entre applications, les coopérations entre communautés
 - nouvelles applications : multimedia, commerce électronique

Un peu d'histoire

– *Database world*

- 1970 SGBDR (relational databases)
- 1990 SGBDRO (nested relational model and object oriented databases)
- 1995 semi-structured databases

– *Documents world*

- 1974 SGML (Structured Generalized Markup Language)
- 1990 HTML (Hypertext Markup Language)
- 1992 URL (Universal Resource Locator)

Data + documents = information

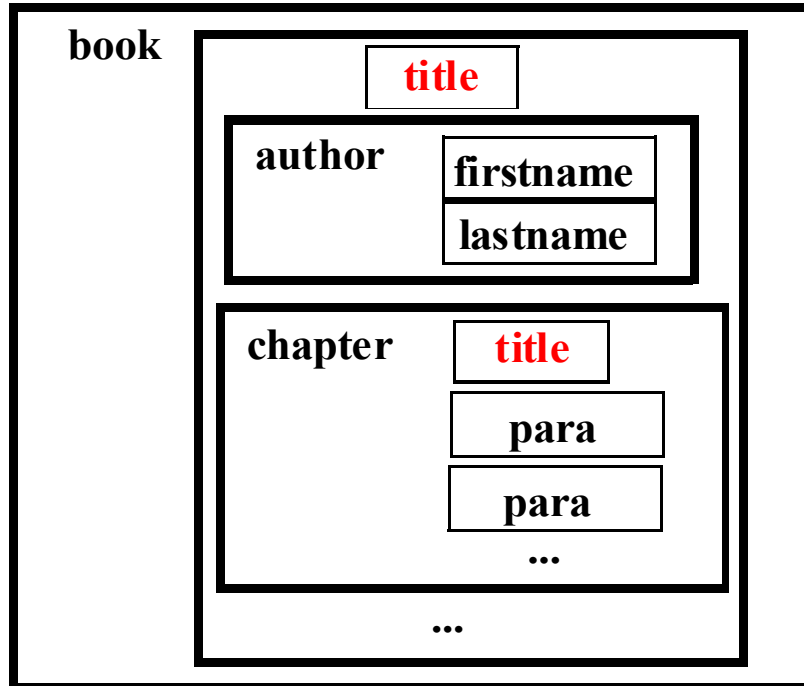
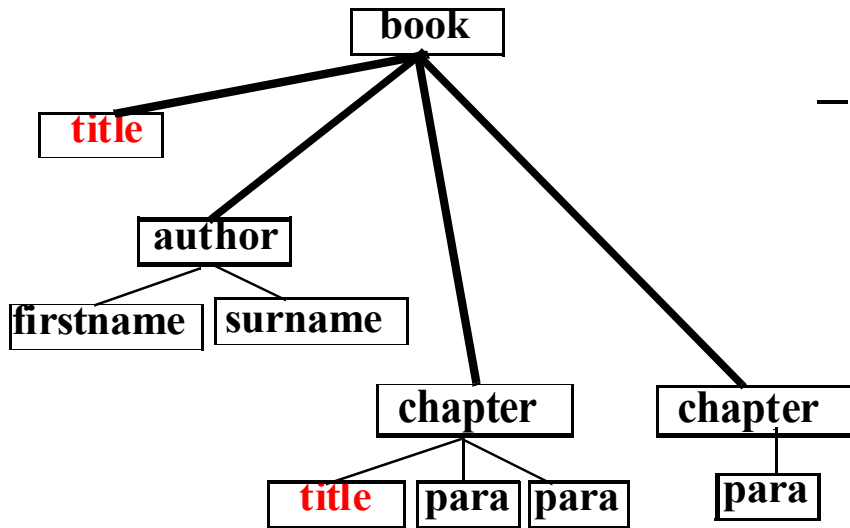
1996 **XML** (Extended Markup Language)

URI (Universal Resource Identifier)

Xml et le document structuré :

- Mécanisme universel
- Utilisable par l'ordinateur
- Lisible par l'homme
- Grande pérennité
- Une clé pour l'interopérabilité
- Méta-langage pour définir d'autres langages
- Un modèle de données semi-structurées
- Une syntaxe d'échange
- L'ASCII du Web
- Beaucoup de bonnes idées de l'informatique réinventées pour le mass-computing

- Un exemple de document balisé
:



```
<book>
  <title>XML langage et
    applications</title>
  <author>
    <firstname>Alain</firstname>
    <surname>Michard</surname>
  </author>
  <chapter>
    <title>Introduction</title>
    <para> bla bla bla </para>
  </chapter>
  ...
</book>
```

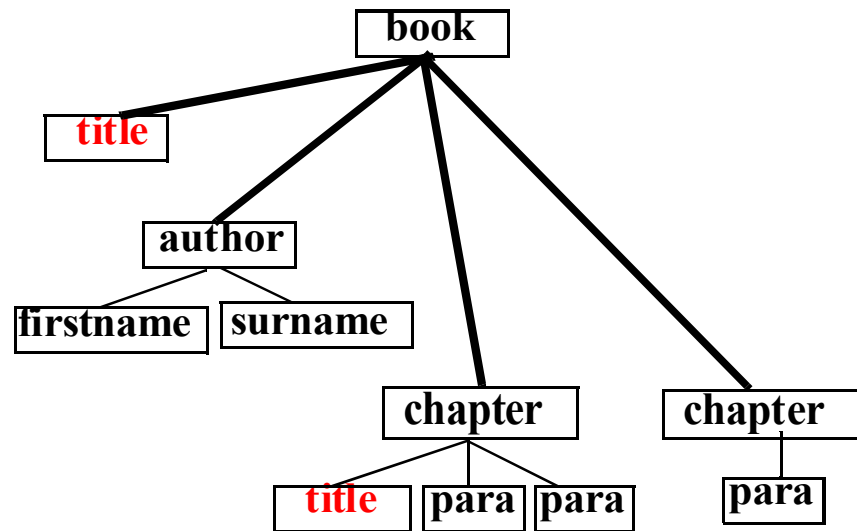
Xml et sa DTD

- DTD : Document Type Definition
 - Définition logique ,
 - et syntaxique du document : Grammaire
- Pourquoi :
 - Échanges contractuels
 - Exemple : Avionneurs vers Compagnies Aériennes
 - Pilotage des éditeurs de texte
 - Pour définir la *validité* d'un document

```

<!DOCTYPE book [
<!ELEMENT book      - - (title,
                          author*,
                          chapter+)>
<!ELEMENT title     - - (#PCDATA)>
<!ELEMENT author    - - (firstname?,
                          surname)>
<!ELEMENT firstname - - (#PCDATA)>
<!ELEMENT surname   - - (#PCDATA)>
<!ELEMENT chap      - - (title?,
                          para+)>
<!ELEMENT para      - - (#PCDATA)>
]>

```



* optionnel, multiple
+ obligatoire, multiple
? optionnel, unique

Plein de DTDs en fonction du domaine

Par exemple, TEI : Text Encoding Initiative

- **Objectif**

- Chercher à décrire tous les textes existant...

- **Références**

- home page TEI :

- <http://www.uic.edu/orgs/tei/>

- **Chimie : CML**

- *(Chemical Markup Language) - Open Molecule Foundation*

- **Documents multimedia SMIL**

- *(Synchronized Multimedia Integration Language) - W3C*

- **Edition : ISO 12083, DocBook...**

- **Mathématiques : MathML - W3C**

- **Graphiques : SVG (Scalable Vector Graphics) - W3C**

`<ellipse rx="85" ry="45"/>`

DTD TEI, Ham vision HTML

```
<html>
```

```
<body>
```

```
<h1>ACT I - SCENE I</h1>
```

```
<p><i>Enter Barnardo and Francisco,  
two Sentinels, at several doors</i></p>
```

```
<p><b>BARN</b> : &nbsp;...&nbsp; Who's there?</p>
```

```
<p><b>FRAN</b> : Nay, answer me. Stand and unfold yourself.</p>
```

```
<p><b>BARN</b> : Long live the King!</p>
```

```
<p><b>FRAN</b> : &nbsp;...&nbsp; Barnardo?</p>
```

```
<p><b>BARN</b> : &nbsp;...&nbsp; He.</p>
```

```
<p><b>FRAN</b> : You come most carefully upon your hour.</p>
```

```
<p><b>BARN : Tis now struck twelve. Get thee to bed, Francisco.</b></p>
```

```
</body>
```

```
</html>
```

ACT I - SCENE I

Enter Barnardo and Francisco, two Sentinels, at several doors

BARN : Who's there?

FRAN : Nay, answer me. Stand and unfold yourself.

BARN : Long live the King!

FRAN : Barnardo?

BARN : He.

FRAN : You come most carefully upon your hour.

BARN : Tis now struck twelve. Get thee to bed,
Francisco.

TEI, Ha vision

ACT I - SCENE I

Enter Barnardo and Francisco, two Sentinels, at several doors

BARN : Who's there?

FRAN : Nay, answer me. Stand and unfold yourself.

BARN : Long live the King!

FRAN : Barnardo?

BARN : He.

FRAN : You come most carefully upon your hour.

BARN : 'Tis now struck twelve. Get thee to bed, Francisco.

```
<div1 type="Act" n="I"><head>ACT I</head>  
<div2 type="Scene" n="1"><head>SCENE I</head>  
<stage rend=italic>Enter Barnardo and Francisco,  
                  two Sentinels, at several doors</stage>  
<sp><speaker>Barn</speaker><l part=Y>Who's there?</l></sp>  
<sp><speaker>Fran</speaker>  
    <l>Nay, answer me. Stand and unfold yourself.</l></sp>  
<sp><speaker>Barn</speaker>  
    <l part=i>Long live the King!</l></sp>  
<sp><speaker>Fran</speaker><l part=m>Barnardo?</l></sp>  
<sp><speaker>Barn</speaker><l part=f>He.</l></sp>  
<sp><speaker>Fran</speaker>  
    <l>You come most carefully upon your hour.</l></sp>  
<sp><speaker>Barn</speaker>  
    <l>'Tis now struck twelve.  
    Get thee to bed, Francisco.</l></sp>  
... </div2> .....</div1>
```

Pleins de perspectives pour le XML :

- **Communauté Document : décrire le contexte, la structure et la sémantique de textes narratifs (blogs, rss etc...)**
- **Communauté Bases de Données : en tant que modèle de données semi-structurées, il capture tout le spectre des données : de « highly structured, regular » à « unstructured »
(relational, object-oriented, HTML, marked up text, ...)**

Pleins d'acronymes pour une famille de technologies :

- XML (Extensible Markup Language)
- XML Namespaces
- XML DTDs, XML Schema
- RDF (Resource Description Framework)

- XSL (Extensible Style Sheet Language)

- XPath (=XSLT \cap XPointer), XLink
- XQL, XML-QL (XML Query Language), Quilt
- XMAS (XML Matching And Structuring language)
- eXcelon, ...
- API Saxe, DOM

On ajoute de la structure (syntaxe) et du sens (sémantique) : **IA**

- XML Document Type Definitions (DTDs):

Définir la structure des documents “autorisés”
(i.e., *valide wrt. une DTD*)

≈ database schema

=> améliorer la formulation des requêtes, leur exécution, ...

- XML Schema

- Définir la structure (DTD) et les data types

- XML Namespaces

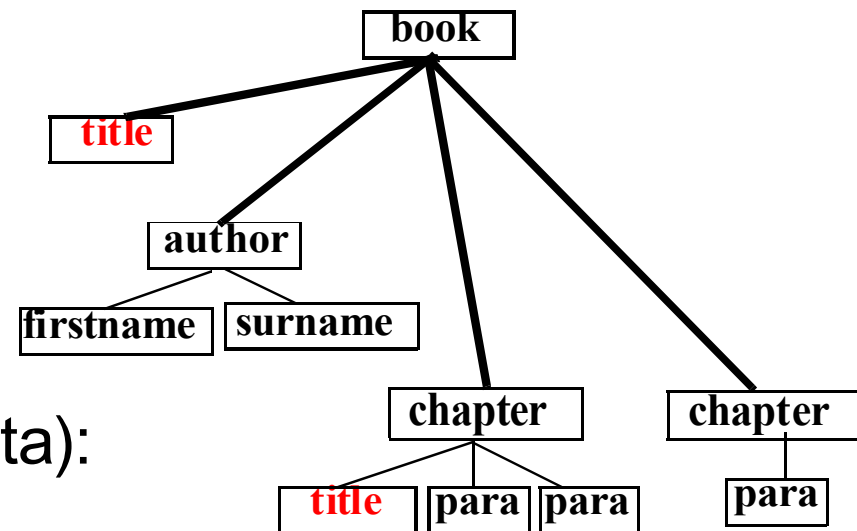
- Identifie le vocabulaire : *svg:set* ou *mathml:set*

“ un cheval bon-marché est rare; un cheval rare est cher; donc un cheval bon-marché est cher » devient “ un cheval bon-marché est *quantite:rare*; un cheval *qualite:rare* est cher »
et pas de fausse interprétation : une des difficultés de l'IA résolue “facilement »

```
<h:html xmlns:xdc="http://.xml.com/books"
        xmlns:h="http://www.w3.org/HTML/1998/html4">
  <h:head><h:title>Book Review</h:title></h:head>
  ...
  <xdc:bookreview>
    <xdc:title>XML: A Primer</xdc:title>
  ...
</h:html>
```

Xml et les API Dom et SAX d'analyse

- **Non-validating parser:**
 - checks that XML doc is syntactically **well-formed**
- **Validating parser:**
 - checks that XML doc is also **valid w.r.t. a given DTD or Schema**
- Parsing yields **tree/object representation:**
 - **Document Object Model (DOM) API**
- Or a **stream of events** (open/close tag, data):
 - Simple API for XML (**SAX**)



XML : Applications & Initiatives industrielles

<http://www.oasis-open.org/cover/xml.html#applications>

- Advertising: **adXML** place an ad onto an ad network or to a single vendor
- Literature: **Gutenberg** convert the world's great literature into XML
- Directories: **dirXML** Novell's Directory Services Markup Language (**DSML**)
- Web Servers: **apacheXML** parsers, XSL, web publishing
- Travel: **openTravel** information for airlines, hotels, and car rental places
- News: **NewsML** creation, transfer and delivery of news
- Human Resources: **XML-HR** standardization of HR/electronic recruiting XML definitions
- International Dvt: **IDML** improve the mgt. and exchange of info. for sustainable development
- Voice: **VoxML** markup language for voice applications
- Wireless: **WAP (Wireless Application Protocol)** wireless devices on the World Wide Web
- Weather: **OMF Weather Observation Markup Format (simulation)**
- Geospatial: **ANZMETA** distributed national directory for land information
- Banking: **MBA Mortgage Bankers Association of America** → credit report, loan file, underwriting...
- Healthcare: **HL7 DTDs** for prescriptions, policies & procedures, clinical trials
- Math: **MathML (Mathematical Markup Language)**
- Surveys: **XML (Data Documentation Initiative)** "codebooks" in the social and behavioral sciences

XML : Initiatives dans le E-commerce

- CommerceNet
 - **eCo Framework** XML specs. to support interoperability among e-businesses
 - **Commerce One** Common Business Library (CBL): set of business components, docs. In DTD, XDR, SOX
 - **BizTalk** Microsoft spec. based on XML schemas
 - **cXML (Commerce XML)** – tag-sets for e-procurement into BizTalk
- Electronic Data Interchange (EDI)
 - **RosettaNet** Common format for online ordering
 - **FpML (Financial products Markup Language)**: sharing of financial data (interest rate & foreign exchange products)
- Open Buying on the Internet (OBI)
 - **OBI** high volume b2b purchasing transactions over the Internet (Office Depot, Lockheed, barnesandnoble, AX..)
- E-commerce and XML
 - **VISA Invoices** The Visa Extensible Markup Language (XML) Invoice Specification provides a comprehensive list of data elements contained in most invoices, including: **Buyer/Supplier, Shipping, Tax, Payment, Currency, Discount, and Line Item Detail.**
- B2B Integration
 - **code360** XML-Broker is middleware software that manages XML based transactions
 - **Bluestone XML Suite** Enables to develop and deploy e-commerce, electronic data interchange, application integration and supply chain management applications. Bluestone XML Suite products include: XML-Server, Visual-XML, XML-Contact and XwingML.
 - **webMethods** Provides companies with integrated direct links to buyers and suppliers

Séparer le contenu (XML) de la présentation (XSLT)

Cascading Style Sheets : CSS comme en HTML

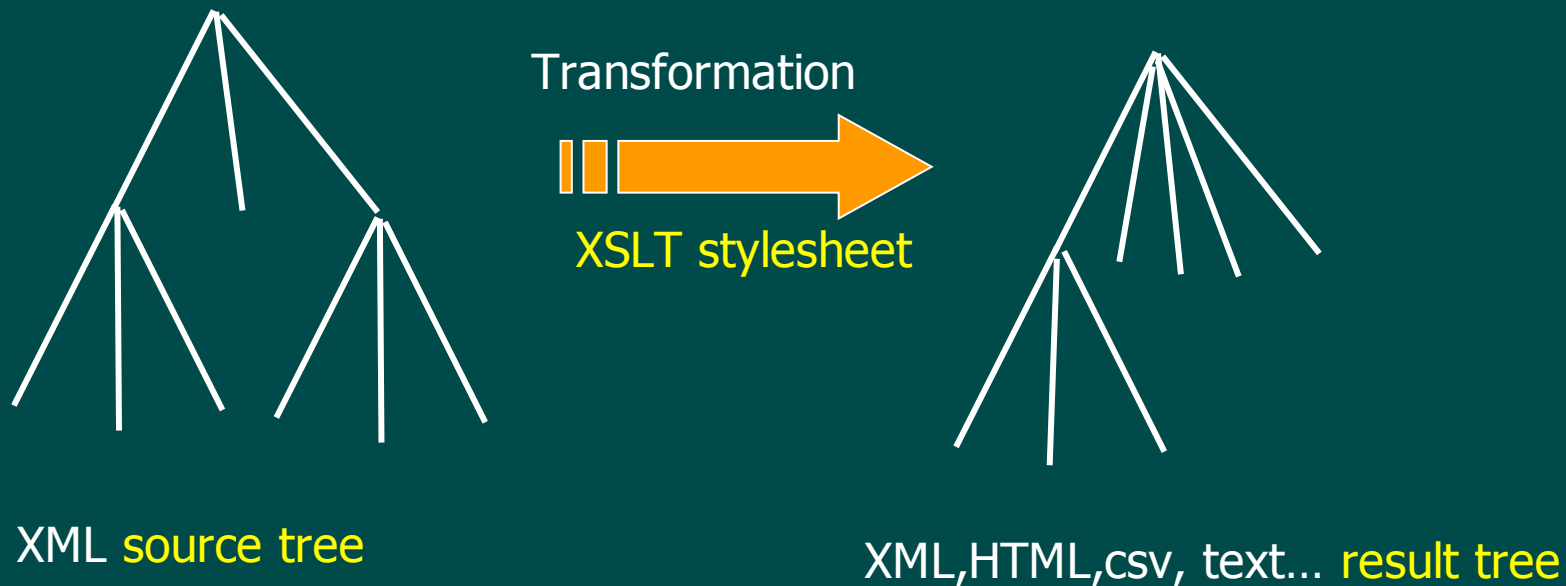
- Attacher des instructions de mises en pages directement aux fichiers XML
 - `<?xml-stylesheet href="http:..." type="text/css" ?>`
 - Supporté par les browsers récents : IE5+, Mozilla, Opera
- Peut mettre en forme (mais pas réarranger les elements)
 - style
 - Bold, italic, underline, font, color, etc.
 - Marges, positionnements

XSL est plus puissant que CSS pour sélectionner les éléments, transformer l'arbre XML, modifier l'affichage en fonction du contenu, des valeurs

- Langage pour transformer les documents XML :
 - en HTML, Text, ou autres documents XMLGénéralement côté serveur

- Interopérabilité et réutilisabilité !

XSLT Processing Model



```
<xsl:template match='//author'>
  <xsl:element name='dc:creator'>
    <xsl:value-of select='lastname' />
    <xsl:text>, </xsl:text>
    <xsl:value-of
      select='firstname' />
  </xsl:element >
</xsl:template>
```


Elements de XSLT ...

- `<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">`
 - root element of an XSLT stylesheet "program"
- `<xsl:template match=pattern name=qname priority=number mode=qname>`
`...template...`
`</xsl:template>`
 - declares a rule: (*pattern* => *template*)
- `<xsl:apply-templates select = node-set-expression mode = qname>`
 - apply templates to selected children (default=all)
 - optional mode attribute
- `<xsl:call-template name=qname>`

pattern

Template Rule: Example

```
<xsl:template match="product">
  <table>
    <xsl:apply-templates select="sales/domestic"/>
  </table>
  <table>
    <xsl:apply-templates select="sales/foreign"/>
  </table>
</xsl:template>
```

template

- (i) **match pattern**: process `<product>` elements
- (ii) **instantiate template**: replace each product element with two HTML tables
- (iii) **select** the `<product>` grandchildren ("`sales/domestic`", "`sales/foreign`") for further processing

Source: *XSLT Programmer's Reference*, Michael Kay, WROX

- take some XML file on books: [books.xml](#)
- now prepare it with style: [books.xml](#)
- and enjoy the result: [books.html](#)
- the recipe for cooking this was: `java com.icl.saxon.StyleSheet books.xml books.xml > books.html`
- and now some different flavors: [books2.xml](#) [books3.xml](#)

The image shows three overlapping windows from Microsoft Internet Explorer. The top-left window displays the XML source file `books.xml`. The top-right window displays the XSLT stylesheet `books.xml`. The bottom window shows the rendered HTML output, titled "A list of books", which is a table listing four books with their authors, titles, and prices.

```
<?xml version="1.0" ?>
- <books>
- <book category="reference">
  <author>Nigel Rees</author>
  <title>Sayings of the Century</title>
  <price>8.95</price>
</book>
- <book category="fiction">
  <author>Evelyn Waugh</author>
  <title>Sword of Honour</title>
  <price>12.99</price>
</book>
- <book category="fiction">
  <author>Herman Melville</author>
  <title>Moby Dick</title>
  <price>8.99</price>
</book>
+ <book category="fiction">
</books>
```

```
<?xml version="1.0" ?>
- <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
- <xsl:template match="books">
  - <html>
    - <body>
      <h1>A list of books</h1>
      - <table width="640">
        <xsl:apply-templates />
      </table>
    </body>
  </html>
</xsl:template>
- <xsl:template match="book">
  - <tr>
    - <td>
      <xsl:number />
    </td>
    <xsl:apply-templates />
  </tr>
</xsl:template>
- <xsl:template match="author | title | price">
  - <td>
    <xsl:value-of select="." />
  </td>
</xsl:template>
</xsl:stylesheet>
```

A list of books

1	Nigel Rees	Sayings of the Century	8.95
2	Evelyn Waugh	Sword of Honour	12.99
3	Herman Melville	Moby Dick	8.99
4	J. R. R. Tolkien	The Lord of the Rings	22.99

XML & databases

- Lien naturel
- Existence d'un langage d'interrogation XML XQuery Language pour faciliter les échanges entre le web et les bases de données (reposant sur Xpath)
- XQuery : un SQL purement logique (pas de gestion d'accès et de stockage)

E.g., List books published by Addison-Wesley after 1991:

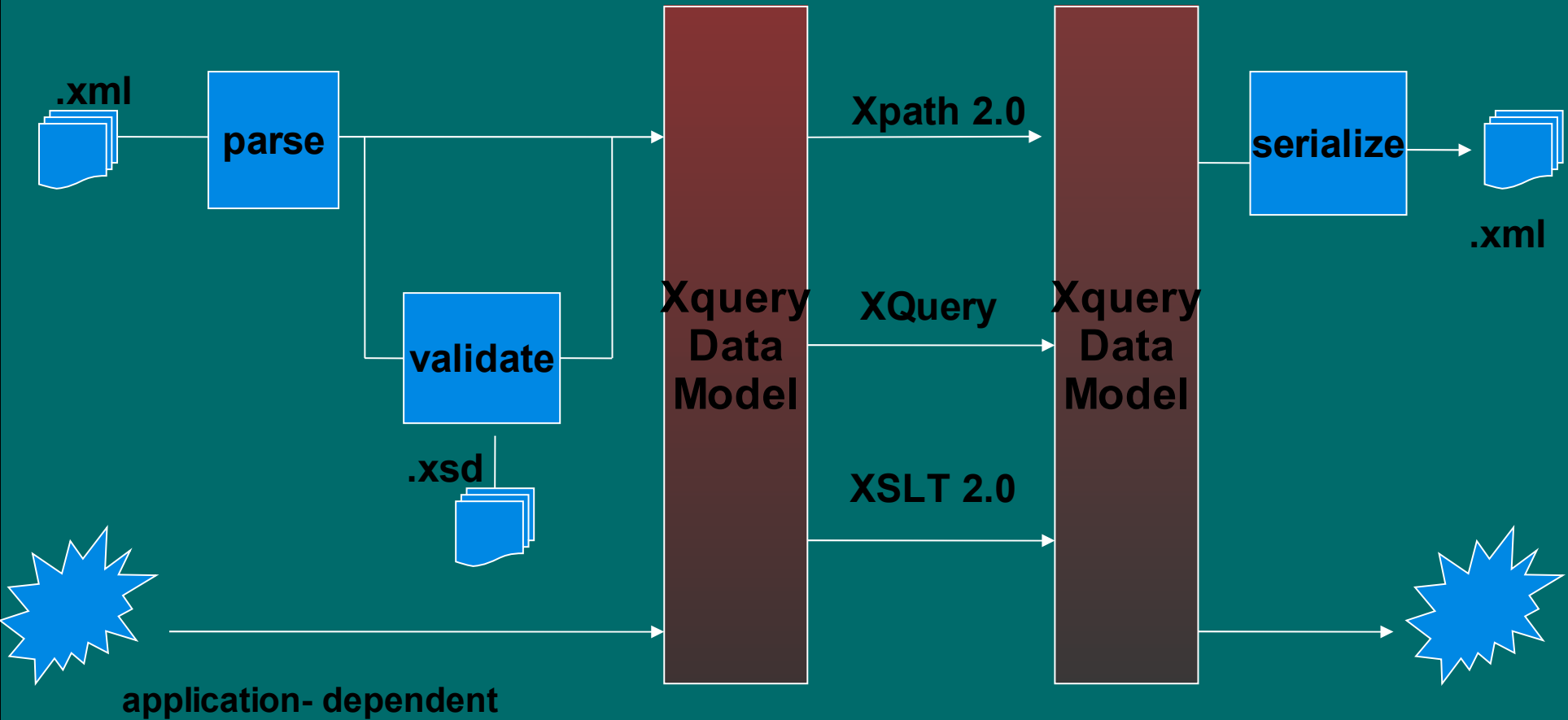
XQuery:

```
<bib> { for $b in document("http://www.bn.com")/bib/book
where $b/publisher = "Addison-Wesley"
and $b/@year > 1991
return <book year="{ $b/@year }"> { $b/title } </book> } </bib>
```

Result:

```
<bib>
  <book year="1994">
    <title>TCP/IP Illustrated</title>
  </book>
  <book year="1992">
    <title>Advanced Programming in the Unix environment</title>
  </book>
</bib>
```

XML Data model life cycle



Exemple: Relational Data vers XML

R

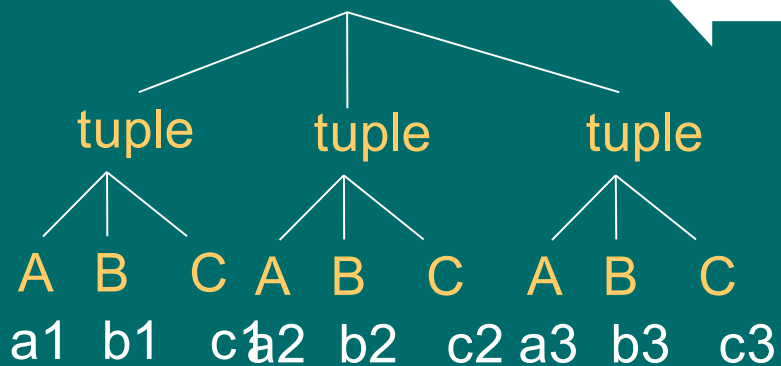
A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3



<R>

```
<tuple>
  <A> a1 </A>
  <B> b1 </B>
  <C> c1 </C>
</tuple>
<tuple>
  <A> a2 </A>
  <B> b2 </B>
  <C> c2 </C>
</tuple>
...
</R>
```

R



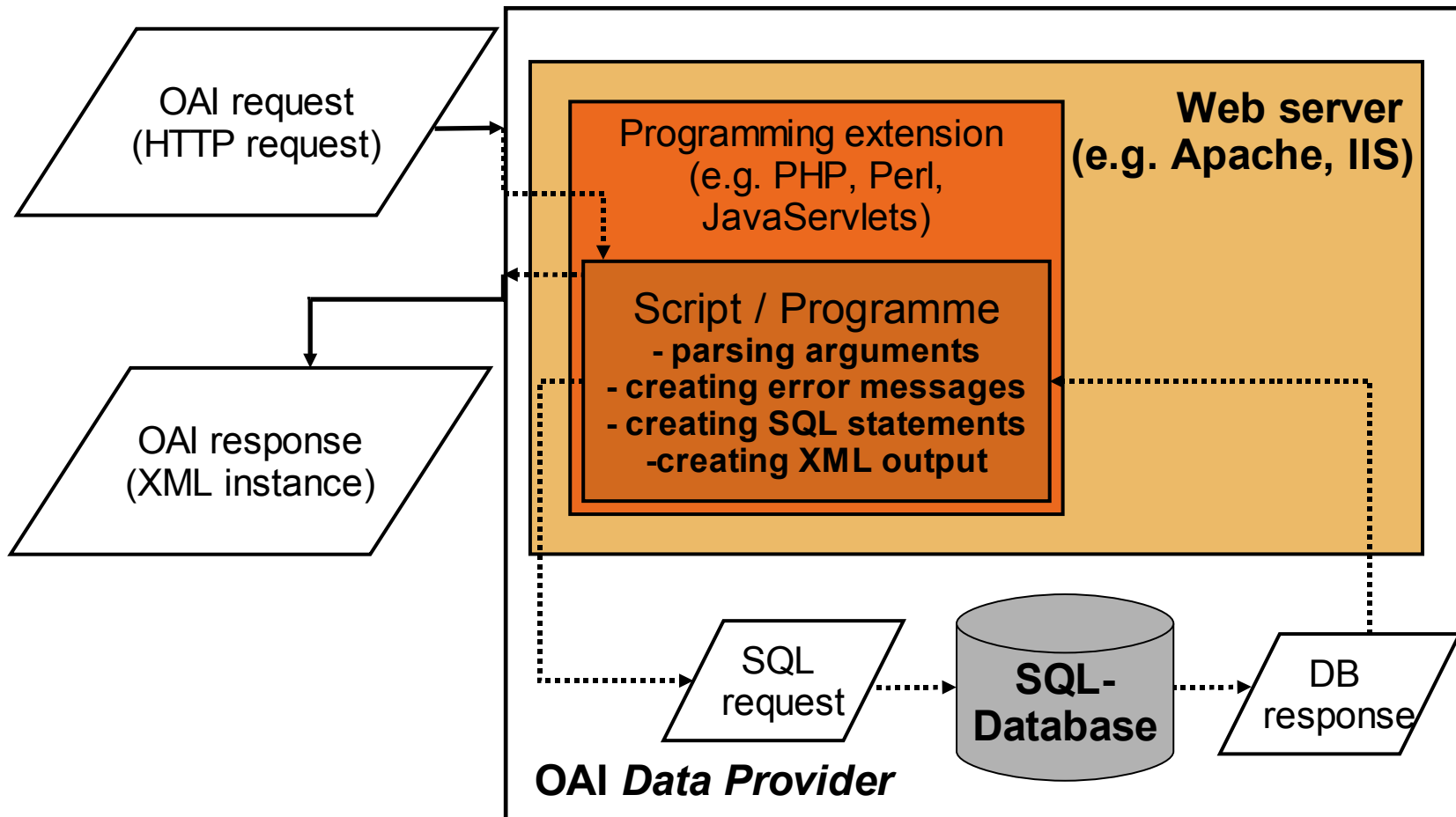
Querying XML

- Different XML QL paradigms depending on the community:
 - (relational, oo, semistructured) **database perspective**
 - Lorel, YaTL, XML-QL, XMAS, FLORA/FLORID, ...
 - **document processing perspective**
 - XQL, XSL(T), XPath, ...
 - **functional programming perspective**
 - QLs with structural recursion, ...
- **APIs Java : DOM** (Object-Oriented approach to traverse the XML node tree, Automatic processing of XML docs, Operations for manipulating XML tree, Manipulation & Updating of XML on client & server, Database interoperability mechanism, Memory-intensive) et **SAX** (The whole file doesn't need to be loaded into memory, XML stream processing, Simple and fast, Allows you to ignore less interesting data but limited expressive power (query/update) when working on streams)

XML & Relational DBMS

- **SQLX**
 - Mapping data types, character sets, ... from SQL to XML
 - Searching XML using XPath-based SQL style queries
- Import / Export of XML from relational database management systems now common
 - Specifics vary by implementation
- **Microsoft implementation** allows:
 - Query relational DB using XPath
 - Query with SQL, return XML
 - Load XML directly into Microsoft SQL
 - OLEDB provider for XML files

Data Provider: Architecture



Creating a simple RSS XML feed using PHP

```
#!/usr/bin/php <?
$DBserver = "localhost";
$DBname = "yourDB";
$DBuser = "username";
$DBpassword = "password";
$webroot = "http://www.yourdomain.com/";
$rootpath = "/var/www/www.yourdomain.com/htdocs/";
$docRoot = $rootpath;
if(!($connect = mysql_connect($DBserver, $DBuser, $DBpassword)))
{ echo "Error Connecting to the Database."; exit(); }

$DB = mysql_select_db($DBname);
$sql = "SELECT BlogDate, BlogTitle, BlogBody, ID FROM tblBlog WHERE Active = 1 ORDER BY BlogDate DESC LIMIT 10";
if(!$rs = mysql_query($sql,$connect))
{ echo "Database Error\n"; echo mysql_error()."\n".$sql."\n"; exit(); }
$rssString = ""; $rssString .= "<?xml version=\"1.0\" encoding=\"utf-8\"?>\n";
$rssString .= "<rss version=\"2.0\">\n"; $rssString .= "<channel>\n";
$rssString .= "<title>YourDomain.com - RSS Feed</title>\n";
$rssString .= "<link>http://www.yourdomain.com/</link>\n";
$rssString .= "<description>YourDomain.com</description>\n";
$rssString .= "<lastBuildDate>".date("D, j M Y H:i:s T")."</lastBuildDate>\n";
$rssString .= "<language>en-us</language>\n";
while($row = mysql_fetch_object($rs)){
    $BlogDate = $row->BlogDate;
    $BlogTitle = htmlspecialchars(strip_tags($row->BlogTitle));
    $BlogBody = substr(htmlspecialchars(strip_tags($row->BlogBody)),0,200)."...";
    $ID = $row->ID;
    $rssString .= "<item>\n";
    $rssString .= "\t <title>".$BlogTitle."</title>\n";
    $rssString .= "\t <link>".$webroot."blog.php?ID=".$ID."</link>\n";
    $rssString .= "\t <guid>".$webroot."blog.php?ID=".$ID."</guid>\n";
    $rssString .= "\t <description>".$BlogDate." - ".$BlogBody."</description>\n";
    $rssString .= "</item>\n";
}
mysql_free_result($rs);
$rssString .= "</channel>\n";
$rssString .= "</rss>\n";
if(!$fp = fopen($rootpath."rss.xml", 'w')){ echo "Could not open the rss.xml file"; exit(); }
if(!fwrite($fp,$rssString)){ echo "Could not write to rss.xml file"; exit(); } fclose($fp);

?>
```