S4 : OMGL1 Module
Advanced Databases for Complex Data Processing

XML

eXtended

Markup

Language

M. Boughanem

Outline of this teaching

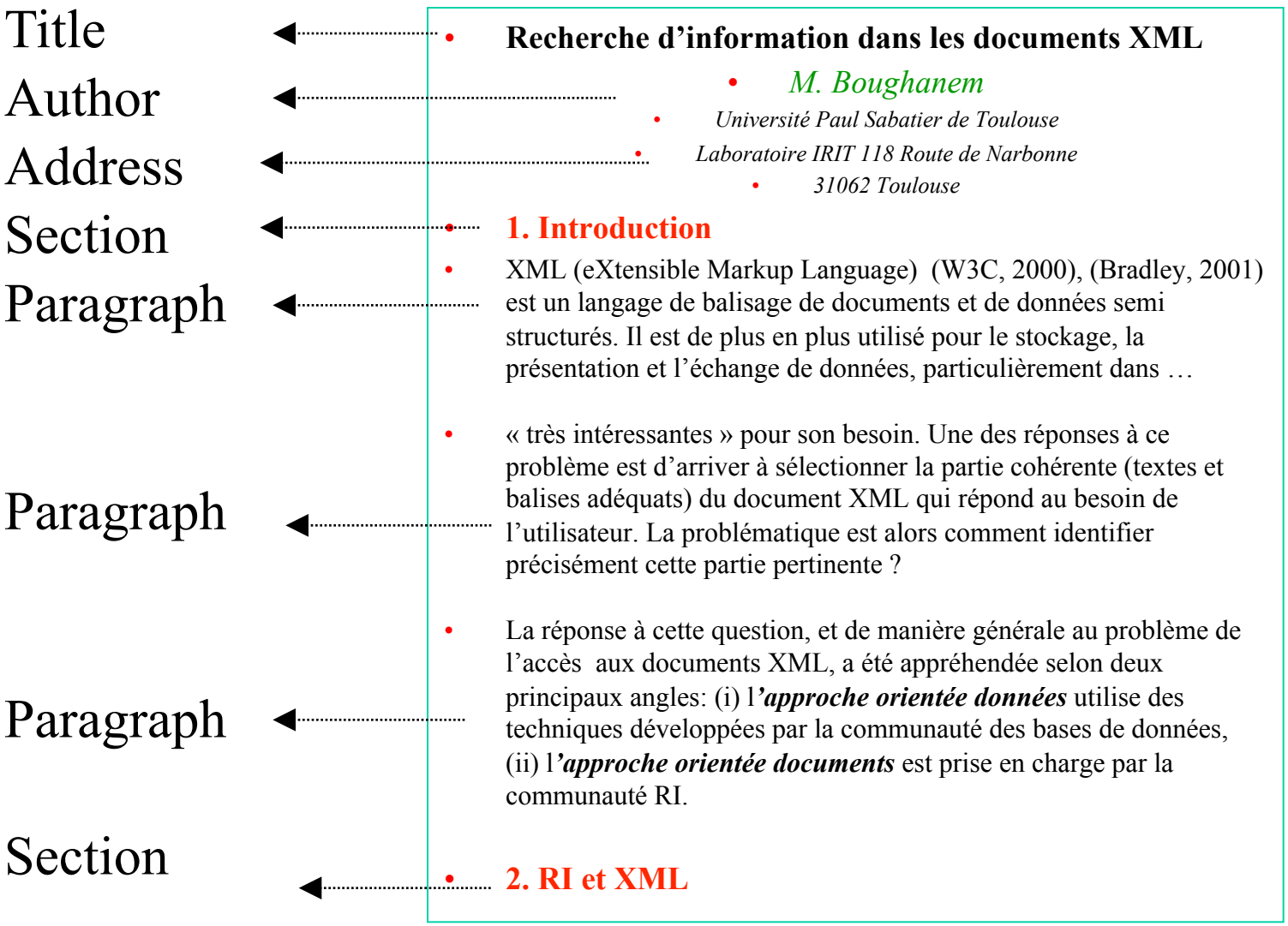
- Lectures in English
  - Lecturer :
    - M. Boughanem
- Tutorials and Labs in English
  - Software: XML Cooktop
  - Lecturers :
    - M. Boughanem & G. Cabanac

Outline

- Part 1: Introduction to XML
- Part 2: Structure of an XML document
- Part 3: Document Type Definition (DTD)
- Part 4: XPath

Part 1
Introduction to XML

Document


Title
Author
Address
Section
Paragraph
Paragraph
Paragraph
Section
Recherche d'information dans les documents XML
M. Boughanem
Université Paul Sabatier de Toulouse
Laboratoire IRIT 118 Route de Narbonne
31062 Toulouse
1. Introduction
XML (eXtensible Markup Language) (W3C, 2000), (Bradley, 2001) est un langage de balisage de documents et de données semi structurés. Il est de plus en plus utilisé pour le stockage, la présentation et l'échange de données, particulièrement dans …
« très intéressantes » pour son besoin. Une des réponses à ce problème est d'arriver à sélectionner la partie cohérente (textes et balises adéquats) du document XML qui répond au besoin de l'utilisateur. La problématique est alors comment identifier précisément cette partie pertinente ?
La réponse à cette question, et de manière générale au problème de l'accès aux documents XML, a été appréhendée selon deux principaux angles: (i) l'approche orientée données utilise des techniques développées par la communauté des bases de données, (ii) l'approche orientée documents est prise en charge par la communauté RI.
2. RI et XML

How to edit a file?

- Microsoft Word exemple.doc
- Microsoft Notepad (or any other file editor)
  ..\exemple\exemple.txt

- … ?
  - Sharing document is _____ due to _____ formats
  - …

The concept of _____ (and markup languages) was invented to distinguish between a document's _____ and _____

XML Document = ____ + _____

- <document>
- <title> Recherche d'information dans des documents XML </title>
- <author> <green> M. Boughanem </green> </author>
- <address> Université Paul Sabatier de Toulouse Laboratoire IRIT, 118 Route de Narbonne
- <zipCode> 31062 </zipCode> <city> Toulouse </city> </address>
- <section title= "Introduction" >
- <par> XML (eXtensible Markup Language) (W3C, 2000), (Bradley, 2001) est un langage de balisage de documents et de données semi structurés. … qui retournent le document entier, en réponse à une requête utilisateur, ne sont plus adéquates. </par>
- <par> En effet, dans le cas particulier d'un document long, la réponse recherchée par l'utilisateur pourrait être " noyée " au milieu d'autres sujets. … identifier précisément cette partie pertinente ? </par>
- <par> La réponse à cette question, et de manière générale au problème de l'accès aux documents XML, a été appréhendée selon deux principaux angles: (i) l'approche orientée données utilise des techniques développées par la communauté des bases de données, (ii) l'approche orientée documents est prise en charge par la communauté RI. <par>
- </section>
- <section title= "RI et XML" > … </section>
- </document>

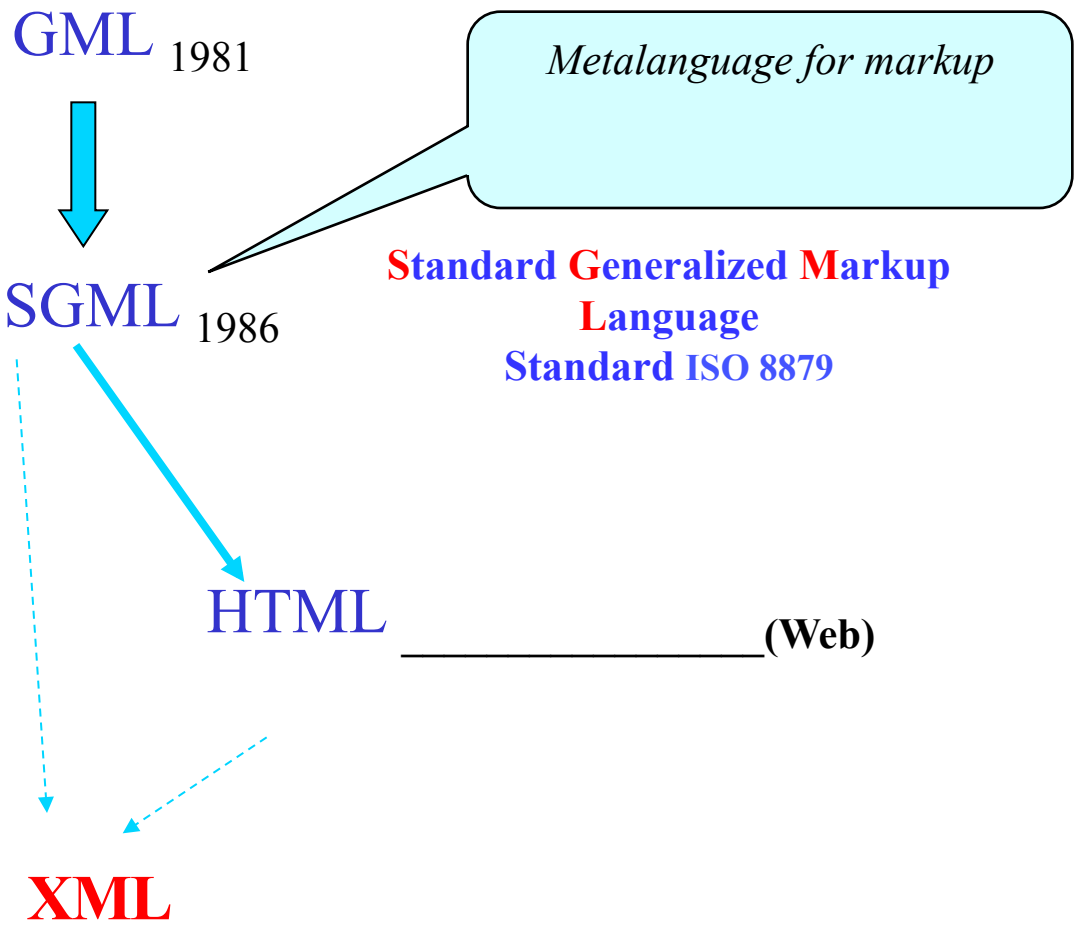
_____ with Tags… Not So New to You

- HTML pages
  - Displayed in an HTML browser (Firefox, Internet Explorer)
  - Displayed in Notepad exemple\exemple_balisé.txt

- HTML (Hypertext Markup Language)
  - _____based on a predefined set of tags (mostly for presentation purposes). Specified in a standard (HTML 1.0, …, 5.0).
  - Tag Semantics:
    - `h1`, .., `h6`, `title`, `address`, … structural tags
    - `center`, `hr`, `b`, `i`, `big`, `small`,… formatting tags

XML = design a document with its own tags

That's a Long Story…



XML 1.0 -10/02/1998

Focus on XML

- XML is an _____
  - "Language", universal "format" to describe the _____ of documents
  - A simplified version of SGML (ISO 8879)

- No predefined "tag set". XML allows document designers to define
  - Their own _____ (with tags)
  - How these tags are organized thanks to the DTD

- A data-model based on the _____structure

Remarks

- XML supplied a _____,
  but no *a priori* _____
- Tags are not related with any formatting or specific _____ in XML, although client applications may well define how to use them
  - <name>Georges</name>
  - <subject>Georges</subject>
- XML defines document _____ only. There is no consideration about how documents are processed by applications.

Remarks

- XML is developed and supported by the W3C
  - Industry: all the big companies, especially Oracle, IBM, Compaq, Xerox, Microsoft, and so on.
  - Research labs: MIT (representing the USA), INRIA (Europe), Keio University (Japan)
    - More details about the W3C World Wide Web Consortium.htm
- XML is expected to _____ the way information is:
  - Shared (XML)
  - Personalized (XSL)
  - _____(XQuery)
  - Secured (Encryption, Signature)
  - Linked (XLink)
  - ...

Advantages of XML for Storing and Sharing Data

- Using XML, any community of authors may freely invent the tags that would store the information they wish to _____

- Example: various ways to store a date
  - `<date>5 January 2000</date>`
  -
```
<date>
  <y>2000</y><m>01</m><d>05</d>
</date>
```
  - `<date format="ISO-8601">2000-01-05</date>`

Advantages: _____

- Each user is free to _____ his/her own document structures
- He/she also can use specific document types, called ____
- Each community can thus propose its own _____ document types
- Validating an XML document against a DTD allows developers to automate _____. This also enables developers to validate data.

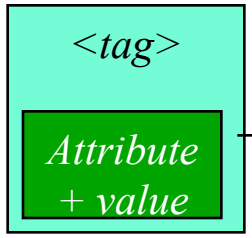
Advantages : Access to heterogeneous data sources

- Retrieving and sharing data among information systems is a _____ task
- XML helps to solve these problems
  - Normalized, _____exchange format
- Indexing and retrieving data from large textual libraries
  - Structural information on top of _____ contents

Part 2:
Structuring an XML document

XML document: Definition

```
<book publicationDate="2000">
<title>Search Engines</title>
<author>John Doe</author>
<chapter>
    <titre>Indexing</titre>
    <section number="1">
        <title>Introduction</title>
        <para>With the advent of…</para>
    </section>
    <section number="2">
        <titre>Web Search Engines</titre>
        <para>Yahoo! Was designed as an …</para>
        <para> Google is a full-text search engine…</para>
    </section>
</chapter>
<chapter> …. </chapter>
</book>
```



Document XML
=
_____
=
(_____ + contents)

An element = **<tag>** ***contents*** **</tag>**

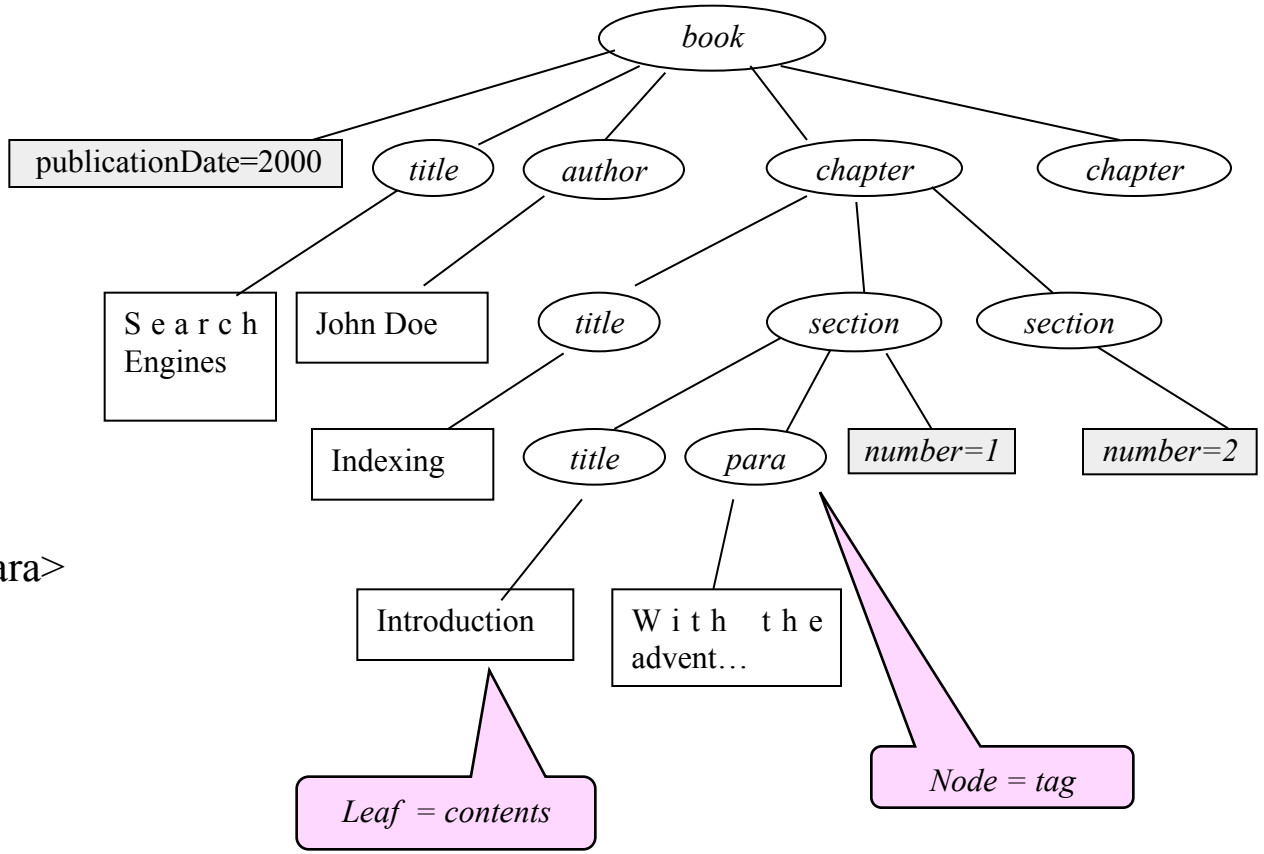
XML Document: Definition

- An XML document is a hierarchical (i.e., _____) data-structure
- Each _____ of the tree has a name
- The rules for structuring an XML document can be stated in a **DTD** (_____)

Visualization

```
….
<!-- root element -->
<book publicationDate="2000">
<!-- children -->
<title>Search Engines</title>
<author>John Doe</author>
<chapter>
      <title>Indexing</title>
      <section number="1" >
         <title>Introduction</title>
         <para>With the adv... </para>
      </section>
      <section number= "2" >…
      </section>
</chapter>

<chapter> …. </chapter>
</book>
```



Structure of an XML Document

- An XML document is composed of:
  - An optional _____

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

  - A *tree of elements (with a _____ element)*

```
<book> <title>Search Engines</title>
        <author>John Doe</author>
        <chapter>
        <section>  <para>With the advent of…</para> </section>
        </chapter>
</book>
```

  - Optional _____ and *processing instructions*

Example: an XML Document

```
<!-- Prologue -->
<?xml  version="1.0" encoding="UTF-8" standalone="yes"?>
<!--DOCTYPE book SYSTEM "doc1.dtd " -->

<!-- root element -->
<book publicationDate="2000">

<!-- First children -->
<title>Search Engines</title>
<author>John Doe</author>
<chapter>
      <title>Indexing</title>
      <section number="1" >
         <titre>Introduction</titre>
         <para>With the advent of…</para>
      </section>
</chapter>
<chapter> …. </chapter>
</book>
```

Contents of the Prologue (1)

- XML Declaration (optional)
  - Version of XML used, name of _____, link to potential external resources

```
<?xml  version="1.0"?>
<?xml  version="1.0" encoding="UTF-8"?>
<?xml  version="1.0" encoding="UTF-8" standalone="no"?>
```

  - *version*: version of XML used in the document, 1.0.
  - *encoding*: then encoding used to store the document. The common encoding used in France _____. The default value for the *encoding* attribute is UTF-8.
  - *standalone*: does the document requires an external DTD for validation?
    - Standalone="yes", independent document, no external DTD.
    - Standalone="no", the _____is informed that the document relies on an external DTD.
    - The default value for the *standalone* attribute is "no".

Contents of the Prologue (2)

- An optional document type declaration

```
<!DOCTYPE book SYSTEM "library.dtd" [ declarations ]>
```

  - Informs the _____ that the document should be validated with the declared DTD

XML Elements

- An XML element is organized as _____(i.e., tree) of elements.
- An XML document as _____in which all other elements are embedded.
- An element is specified by its **name** and **contents**
- Contents lie between a _____and an optional **closing tag**.

Syntax:

- **<ElementName>***contents*</ElementName>**
- **<ElementName/> (empty element)**

Examples:

- **<address>**118, route de Narbonne**</address>**
- **<true/>**

Constraints about Element Naming

- The name of an element/attribute is a character string that may contain:
  - _____, underscore (_), dash (-), full stop (.)

- The element/attribute name must comply with the following rules:
  - The first character is a letter or an underscore
  - The 3 first characters must not be "_____" (upper or lower case)

Example of Element Names	
Right	Wrong
`_foo`	`1998-catalogue`
`Company_name`	`XmlSpecification`
`xsl:rule`	`society name`
`X.11`	

Element Attributes

- All elements may contain any number of attributes
- An attribute is a _____ (name, value)
  - <section number="1">Search Engines</section>
- An attribute can **only** be found in the _____ of its element
  - Example:
  - This is incorrect: </book lang="en">

Entities

- Entities establish a link between a _____and a replacement text or a handle pointing to an external resource.
  Entities may be of two kinds:
  - Predefined entities to be used instead of some characters (<, >, &, ‘, “) that cannot be used in XML as they are part of the markup

Entity	Value	Example	Result
<	less than (<)	10 < 100	10 < 100
>	greater than (>)	x > 119	x > 119
&	ampersand (&)	AT&T	AT&T
'	apostrophe (‘)	d'autres	d’autres
"	quote(")	"Wow!"	"Wow!"

  - Character entities to be used for any character based on its code. These are written &#n; in XML, where n is a number (see ISO 10646 for character codes).
    - Example: *é* can be coded as é
  - One can define his/her own entities. One may also _____from other DTDs. There are global or parametric entities (see the part on DTD).

Comments in XML

- XML comments are written in the same way as in HTML.
- Comments start with _____ and end with _____.
- Comments may be located anywhere in the XML document but they _____.
- Example of valid comments:
  - <!-- this is correct -->
  - <elt> <!-- this is correct as well -->
    Babbles babbles</elt>

CDATA Sections

- ***CDATA*** Sections: _____
  - May contain any kind of contents
  - Defines a block of text that should not be interpreted by the parser
  - Allows the _____ with protected characters

Syntax:

```
<![CDATA[ … contents … ]]>
```

Example:

```
<![CDATA[ IF A<B THEN PRINT A+B ]]>
```

<![CDATA[A start tag starts by a '<' and ends with a '>'.]]>

Exercices

- Design an XML document that stores the contents of a letter
  - Information about the recipient: first name, last name, address (number, street, city, ZIP code, country)
  - Body of the message

Part 3 :
DTD (Document Type Definition)

DTD
(Document Type Definition)

- Declares the vocabulary (i.e., elements with tags and attributes) and the organization of an XML document
- The DTD is the grammar of the language whose sentences are XML documents (instances)
- The DTD can be declared or stored in two different locations:
  - Embedded in the XML document (i.e., *internal DTD*)
  - Stored in a separate file (i.e., *external DTD*).
- When both internal and external DTDs are declared in an XML file, these are merged by the parser so that the _____ DTD adds to the internal DTD
- The DTD declaration is found just _____

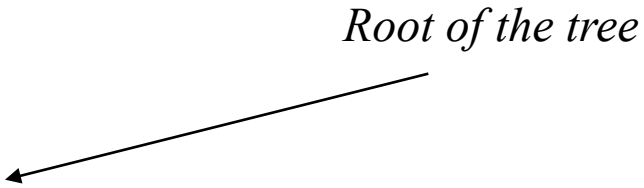
Example of a DTD

```
<!DOCTYPE book [
<!-- The DTD is just below -->

<!ELEMENT book ( title, author, chapter+)>
<!ELEMENT chapter (title, section*)>
<!ELEMENT section (title, para+)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ATTLIST book publicationDate CDATA #REQUIRED>
...
]>
```

Declaring an internal DTD

- The declaration is stored in the XML document
- Internal DTD




```
<!DOCTYPE book [
Definition of the elements
..
]>
<book> …..</book>
```

Example of an internal DTD

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<!DOCTYPE book [
<!- The DTD is just below -->
<!ELEMENT book (title, author, chapter+)>
<!ELEMENT chapter (title, section*)>
<!ELEMENT section (title, para+)>
…]>

<book>
<title>Search Engines</title>
<author>John Doe</author>
<chapter>
    <title>Indexing</title>
    <section>
      <title>Introduction</title>
      <para>With the advent of…</para>
    </section>
</book>
```

Declaration of an external DTD

- External DTDs are either _____ or _____
  - Private when it is _____ on the local file system (on the programmer's computer),
    - Example of declaration:

      <!DOCTYPE book SYSTEM "book.dtd">
  - Public when it is located on the Internet
    - Example of declaration:

      <!DOCTYPE book PUBLIC "http://www.e-xmlmedia.com/doc.dtd">

Designing of a DTD: Elements

- <!ELEMENT Tag elementType >
  - Tag: name of the element,
  - elementType = one of the _____ contents:
    - Textual contents (#PCDATA) [Parsed Character DATA]
    - No contents (EMPTY)
    - A sequence list of elementTypes (,)
    - A choice list of elementTypes (|)

Element types:
Parsed Character Data (#PCDATA)

- **#PCDATA**: *Parsed Character DATA* are text with _____

Syntax: **<!ELEMENT** elementName (**#PCDATA)>**

Example: <!ELEMENT title (#PCDATA)>

XML document: <title>Search Engines</title>

Element types: ANY and EMPTY

- **EMPTY**: the element has _____

Syntax: `<!ELEMENT elementName EMPTY>`

Example: `<!ELEMENT null EMPTY>`

XML document: `<null/>`

- **ANY**: the element has any kind of contents (another **element**, **text**, or **nothing** (empty))

Syntax: `<!ELEMENT elementName ANY>`

Example: `<!ELEMENT whatever ANY>`

XML document: `<whatever> <b1>babbles…</b1><b2/> </whatever>`

Element types: Sequence lists

- A **sequence list of elements** is an ordered list. It contains the _____ of the element that we are currently defining

<!ELEMENT book (title, author, chapter)>

Valid Example

```
<book>
  <title> …</title>
  <author> …</author>
  <chapter> …</chapter>
</book>
```

Not valid example

```
<book>
  <title> …</title>
  <chapter> …</chapter>
</book>
```

The author is missing

Element types:
Choice lists

- A _____ is an unordered list of elements. It defines the possible choices of children for the current element

<!ELEMENT library (book | article)>

Valid examples

<library> <book> …
</book>
</library>

<library> <article> …
</article>
</library>

Incorrect example

<library>
<book> … </book>
<article> … </article>
</library>

Element types: Quantifiers

- _____ may be applied to elements, sequence lists, or choice lists.
  - No quantifier: the element must appear only once
  - **?** the (group of) element(s) must appear once or not at all
  - * the (group of) element(s) must appear zero, one or several times
  - + the (group of) element(s) must appear once or several times

<!ELEMENT book (title, author+, abstract?, chapter*)>

Example <!ELEMENT library (book | article)*>

The following document is now valid

<library>
<book> … </book>
<article> … </article>
</library>

List of Elements

- A _____(i.e., tree-like) is created by listing its various elements. This implies that elements are embedded into other elements (recursive embedding).

```
<!ELEMENT book(title, author, chapter+)>
<!ELEMENT chapter(title, section*)>
<!ELEMENT section(title, para+)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT para (#PCDATA)>
```

```
<book>
<title>Search Engines</title>
<author>John Doe</author>
<chapter>
<title>Indexing</title>
<section>
 <title>Introduction</title>
 <para>With the advent of</para>
</section>
</chapter>
</book>
```

Element types: Mixed elements

- Sequence lists may contain elements and #PCDATA with quantifiers (e.g., *)
- A single level of description
- In an XML document, _____may appear multiple times in any order

Syntax for DTD: **<!ELEMENT** elementName **(#PCDATA** | element | element | **...)*>**

Example of a DTD: <!ELEMENT sentence (#PCDATA | quote | name)*>

XML Document:

```
<sentence>
   As <name>Tim Bray</name> stated <quote>...</quote>
</sentence>
```

Declaring Attributes (1/6)

- Attributes _____ to elements (e.g., link between two elements)

- Declaration
  - <!ATTLIST elementName [attribute type #option ["defaultValue"]]* >

Declaring Attributes (2/6)

- Type of attributes
  - CDATA: character string surrounded by " "
  - Set of values (val1| val2|…)
  - NMTOKEN / NMTOKENS: a valid name token in XML
  - ENTITY / ENTITIES: a _____ that should appear in an <!ENTITY….> declaration
  - ID – IDREF –IDREFS : internal references
    ID: creation IDREF: use
  - NOTATION: symbolic name that points to a <!NOTATION….>

- Options
  - **#REQUIRED** Suppling a value for the attribute is _____
  - **#IMPLIED** Suppling a value for the attribute is _____
  - **#FIXED** Implicit value that is defined *a priori*

Declaring Attributes (3/6)

- Default values
  - Example in DTD
    - <ELEMENT publisher #PCDATA>
    - <!ATTLIST publisher city CDATA "Toronto">
  - Example in XML
    - <publisher city="Paris"> …</publisher>
- _____ values (#IMPLIED)
  - Example in DTD
    - <!ATTLIST publisher city CDATA #IMPLIED >
  - Example in XML
    - <publisher city="Toulouse"> …</publisher>
    - <publisher> …</publisher>

Declaring Attributes (4/6)

- _____ values (#REQUIRED)
  - Example in DTD
    - <!ATTLIST publisher city CDATA #REQUIRED >
  - Example in XML
    - <publisher city="Las Vegas"> …</publisher>
    - <publisher> …</publisher> is not valid
- Fixed values (#FIXED)
  - Example in DTD
    - <!ATTLIST university name CDATA #FIXED "UPS" >
  - Example in XML
    - <university name="UPS"> …</university>
    - <university> …</university> is not valid

Declaring Attributes (5/6)

- _____ attributes (val1 | val2…)
  - Example in DTD
    - <!ATTLIST payment type (VISA | check) "check">
  - Example in XML
    - <payment type="VISA"> …</payment>

- Other examples
  - Example in DTD
    - <!ATTLIST author gender (M | F)
      country CDATA #IMPLIED> (specify a default value)
  - Example in XML
    - <author gender="M" country="USA">Dan Brown</auteur>

Declaring Attributes (6/6)

Example

```
<!DOCTYPE book [
<!-- The DTD is just below -->

<!ELEMENT book (title, author, chapter+)>
<!ELEMENT chapter (title, section*)>
<!ELEMENT section (title, para+)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author(#PCDATA)>
<!ELEMENT para(#PCDATA)>
<!ATTLIST book publicationDate CDATA #REQUIRED>
<!ATTLIST section number CDATA #REQUIRED>
]>
```

Declaring entities

- Entities allow the_____ of:
  - A variable that would be usable throughout the document (**global entity**)
  - An alias only usable within the DTD (**parametric entity**)

- **Internal** entities are declared in the document

- _____ refer to external resources

- There are other types of entities: predefined (e.g., &) and character entities (ڀ) see part 2

Example of an Internal DTD

Stored in a single file

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<!DOCTYPE book [
<!- The DTD is just below -->
<!ELEMENT book (publicationDate, title, author, chapter+)>
<!ELEMENT chapter (title, section*)>
<!ELEMENT section (number, title, para+)>
…]>

<book>
<publicationDate="2000">
<title>Search Engines</title>
<author>John Doe</author>
<chapter>
    <title>Indexing</title>
    <section number= "1”>
      <title>Introduction</title>
      <para>With the advent of…</para>
    </section>
</book>
```

Example of an _____

```
<?XML version="1.0" encoding="UTF-8" ?>
<!DOCTYPE book SYSTEM "book.dtd">

<book publicationDate="2000">
  <title>Search Engines</title>
  <author>John Doe</author>
    <chapter>
      <title>Indexing</title>
        <section number= "1”>
          <title>Introduction</title>
          <para>With the advent of…</para>
        </section>
    </chapter>
    <chapter> ….
    </chapter>
</book>
```

```
<!– The DTD follows -->
<!ELEMENT book (title, author, chapter+)>
<!ELEMENT chapter (title, section*)>
<!ELEMENT section (title, para+)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT para (#PCDATA)>
<!ATTLIST book publicationDate CDATA
      #REQUIRED>
<!ATTLIST section number CDATA #REQUIRED>
```

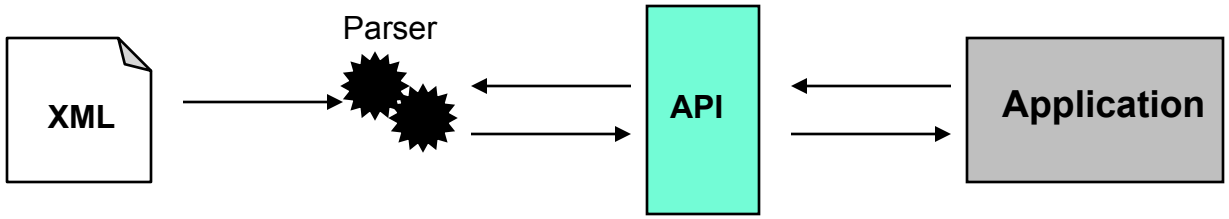
DTD stored in the file book.dtd

Validity of XML Documents

- _____ Document
  - Tags correctly embedded
  - Parsable document
  - Not necessarily valid against the DTD
- _____ Document
  - Well-formed + complying to the DTD

XML Parsers

- An _____ is a software that reads and processes XML documents
  - Checks that a document is weel-formed and valid
  - Extracts the data
- Various types of parsers
  - Validating or non-validating
  - Based on the _____
  - Based on the _____
  - And so on…



The End

Internal Global Entities

- Used to _____ in a document (<=> constants)

Syntax of declaration:

<!ENTITY entityName "*textual contents of the entity*"**>**

Syntax of use in an XML document:

&entityName**;**

Example:

<!ENTITY H "Hydrogen"**>**

<SENTENCE>&H; is the lighter of all elements</SENTENCE>

Demo (as opened in IE):

Hydrogen is the lighter of all elements

External Global Entities

- Used when the _____ is outside of DTD

Syntax of declaration: **<!ENTITY** entityName **SYSTEM** "*resource identifier*"**>**

<!ENTITY entityName **PUBLIC** "*Public Identifier*" "*URI of the Resource*"**>**

Declaration in a DTD:

<!ENTITY Author SYSTEM "author.xml">

<!ENTITY Author PUBLIC "-//EBSI//TEXT Organisme//FR"
"http://www.irit.fr/bougha/author.xml">

Example of use:

author.xml

<?xml version="1.0" encoding="UTF-8"?>
<AUTHOR>Gerard Salton</AUTHOR>

Use in a document:

<SENTENCE>&Author; wrote " IR " </SENTENCE>

Result:

<SENTENCE>Gerard Salton wrote “IR” </SENTENCE>

Entities

- Internal Parameter Entities allow the definition of symbols (or entity types) that can be used in the DTD. They are “shortcuts.”
- May be internal or external.
- May be declared and used only in the DTD.

Syntax for declaring internal entities:

<!ENTITY % entityName "*contents*"**>**

Syntax of utilize:

%entityName**;**

Examples:

```
<!ENTITY % contents “(#PCDATA|em)">
<!ELEMENT p %contents;>

<!ENTITY %  address      "number CDATA # IMPLIED
                          street CDATA #IMPLIED
                          zipCode CDATA #IMPLIED">
<!ATTLIST recipient %address;>
```

External Parameter Entities

- External Parameter Entities use a _____ that refers to a their location.

Syntax for declaring an external parameter entity:

<!ENTITY % entityName **SYSTEM** "*URI refering to the contents of the entity*">

<!ENTITY % entityName **PUBLIC** ”Public Identifier" "*URI refering to the entity*">

Syntax of use:

%entityName**;**

Example:

```
<!DOCTYPE document SYSTEM "mydtd.dtd"
[<!ENTITY % greekChars PUBLIC "greekchars.dtd">

%greekChars;
]>
```