

## BNF-Syntax for book Example

```
<book>      ::= <title> <authors> <publisher1> <sections>
<title>     ::= string
<authors>   ::=      | <authors> <author>
<author>    ::= string
<sections>  ::= <section> | <sections> <section>
<section>   ::= string | <title> | <section>
```

6

## A first look at XML

```
<?XML version="1.0"?>
<!DOCTYPE book [
  <!ELEMENT book (title, author*, publisher?, section+)>
  <!ATTLIST book year CDATA #IMPLIED>
  <!ELEMENT title (#PCDATA)>
  <!ENTITY %macro "publisher (#PCDATA)">
  <!-- The declaration of the <publisher> element-->
  <!ELEMENT %macro;>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT section (#PCDATA | title | section)*> ]>
<book year="1967" >
  <title>The politics of experience</title>
  <author>R.D.Laing</author>
  <section>
    The great and true Amphibian, whose nature is disposed to....
    <title>Persons and experience</title>
    <section> <![CDATA[Exploitation <must> not been....]]> </section>
  </section>
</book>
```

D.Florescu, J. Siméon

Int. Conf. On Very Large Databases, Cairo, 2000

7

## A short XML lexicon

- ◆ **XML Document:**
  - a well-formed data object
  - conforming SGML document
  - made of markup and character data
- ◆ **Well-formed document:**
  - matches the document production in the XML grammar
  - contains properly nested elements with a single root element
- ◆ **Document Type Definition (DTD):**
  - provides a grammar for the document
  - contains or points to markup declarations for: elements, attributes, entities, notations
  - optional
- ◆ **Valid document:**
  - has an associated DTD or Schema
  - complies with the associated constraints contained in it

```
<?XML version="1.0"?>
<!DOCTYPE book [
  <!ELEMENT book (title, author*, publisher?, section*)>
  <!ATTLIST book year CDATA #IMPLIED>
  <!ELEMENT title (#PCDATA)>
  <!ENTITY %macro "&publisher (#PCDATA)">
  <!-- The declaration of the <publisher> element-->
  <!ELEMENT %macro;>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT section (#PCDATA | title | section)*>
]>
<book year="1967">
  <title>The politics of experience</title>
  <author>R.D.Laing</author>
  <section>
    The great and true Amphibian, whose nature is disposed to....
    <title>Persons and experience</title>
    Even facts become fictitious without adequate ways to...
  </section>
  <section><section><![CDATA[Exploitation <must> not been....]]>
    </section>
  </section>
</book>
```

D.Florescu, J. Siméon

Int. Conf. On Very Large Databases, Cairo, 2000

8

## Elements

- ◆ **Element:**
  - the logical atomic unit of data
  - has a name, a content and a set of attributes
  - the content is an ordered list of children that can be elements, character data, comments, processing instructions and references
- ◆ **Element declaration:**
  - describes constraints on the content of an element
  - EMPTY: no content allowed
  - ANY: can contain any elements defined in the DTD, in any order
  - MIXED: character data mixed with the additional declared elements
  - CHILDREN: the children can be only elements and they have to satisfy the given regular expression

```
<?XML version="1.0"?>
<!DOCTYPE book [
  <!ELEMENT book (title, author*, publisher?, section*)>
  <!ATTLIST book year CDATA #IMPLIED>
  <!ELEMENT title (#PCDATA)>
  <!ENTITY %macro "&publisher (#PCDATA)">
  <!-- The declaration of the <publisher> element-->
  <!ELEMENT %macro;>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT section (#PCDATA | title | section)*>
]>
<book year="1967">
  <title>The politics of experience</title>
  <author>R.D.Laing</author>
  <section>
    The great and true Amphibian, whose nature is disposed to....
    <title>Persons and experience</title>
    Even facts become fictitious without adequate ways to...
  </section>
  <section><section><![CDATA[Exploitation <must> not been....]]>
    </section>
  </section>
</book>
```

D.Florescu, J. Siméon

Int. Conf. On Very Large Databases, Cairo, 2000

9

# Attributes

- ◆ Attribute:
  - (name, string value) pair
  - associated with an element
- ◆ Attribute declaration:
  - a triple (name, type, defaultValue)
  - type:
    - » string type (CDATA)
    - » tokenized type (ID, IDREF, IDREFS, entity, nmToken, etc)
    - » enumerated
  - default declaration:
    - » REQUIRED
    - » IMPLIED
    - » FIXED
    - » default value

```
<?XML version="1.0"?>
<!DOCTYPE book [
  <!ELEMENT book (title, author*, publisher?,
  section*)>
  <!ATTLIST book year CDATA #IMPLIED>
  <!ELEMENT title (#PCDATA)>
  <!ENTITY %macro "publisher (#PCDATA)">
  <!-- The declaration of the <publisher> element-->
  <!ELEMENT %macro;>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT section (#PCDATA | title | section)*>
]>
<book year="1967">
  <title>The politics of experience</title>
  <author>R.D.Laing</author>
  <section>
    The great and true Amphibian, whose nature is disposed
    to.... .
    <title>Persons and experience</title>
    Even facts become fictions without adequate ways to...
  </section>
  <section><section><![CDATA[Exploitation <must> not
  been...]]></section>
  </section>
</book>
```

D.Florescu, J. Siméon

Int. Conf. On Very Large Databases, Cairo,2000 10

# Entities

- ◆ Entities:
  - the physical storage unit for the XML data
  - have a name and a content
  - can be referenced by name
  - first classification:
    - » parsed entities
    - » unparsed entities
  - second classification:
    - » internal entities
    - » external entities
  - parsed entities:
    - » general entities: can occur in the data content of the document
    - » parameter entities : can occur in the DTD
  - entity references:
    - » general entity: &name;
    - » parameter entity: %name;

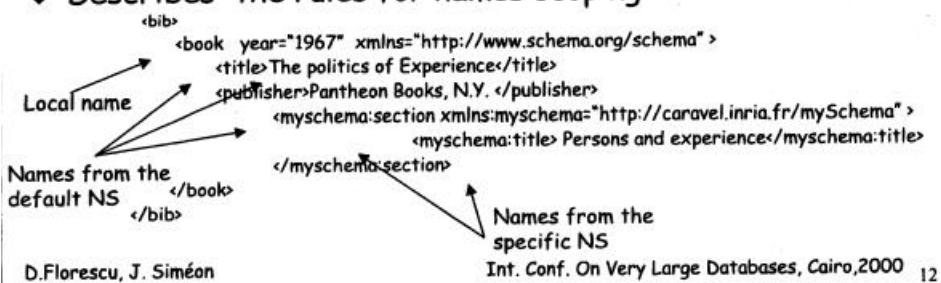
```
<?XML version="1.0"?>
<!DOCTYPE book [
  <!ELEMENT book (title, author*, publisher?,
  section*)>
  <!ATTLIST book year CDATA #IMPLIED>
  <!ELEMENT title (#PCDATA)>
  <!ENTITY %macro "publisher (#PCDATA)">
  <!-- The declaration of the <publisher> element-->
  <!ELEMENT %macro;>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT section (#PCDATA | title | section)*>
  <!ENTITY macro2 "<![CDATA[Exploitation <must>
  not been...]]>">
]>
<book year="1967">
  <title>The politics of experience</title>
  <author>R.D.Laing</author>
  <section>
    The great and true Amphibian, whose nature is disposed
    to.... .
    <title>Persons and experience</title>
    Even facts become fictions without adequate ways to...
  </section>
  <section><section> &macro2 </section></section>
</book>
```

D.Florescu, J. Siméon

Int. Conf. On Very Large Databases, Cairo,2000 11

## Namespaces

- ◆ An XML namespace = a collection of names identified by an Universal Resource Identifier (URI)
- ◆ The names used for tag names and attribute names become qualified names (Qnames)
- ◆ QName = an optional namespace prefix followed by a required local part
- ◆ Describes the rules for names scoping



## XML data model in a nutshell

- ◆ XML documents describe nested tagged elements  
= syntax for trees
- ◆ XML data model describe information content  
= data model for tree-based structures

## Why a data model for XML ?

For old & well-known (but good!) reasons

- ◆ As a support for physical/logical independence
  - > XML can be stored in files, in a native XML repository, in a relation databases
  - > XML can be kept virtual
    - as a view of an underlying repository
    - as a view of integrated data sources
  - > XML can be in a main-memory data structures (in C++, Java, etc)
  - > XML can be streamed between processes
- ◆ To describe information content of XML documents
  - > to agree and reason about information content, preservation
- ◆ To define semantics of operations:
  - > equality, etc.

D.Florescu, J. Siméon

Int. Conf. On Very Large Databases, Cairo,2000 14

## XPath

- ◆ Papers:
  - "XML Path Language (Xpath)", W3C recommendation
- ◆ Used as a building block for:
  - XSL Transformations (XSLT)
  - XML Pointer (Xpointer)
  - XML Link (Xpointer)
- ◆ Syntax for tree navigation and node selection

D.Florescu, J. Siméon

Int. Conf. On Very Large Databases, Cairo,2000 95

## XPath design

- ◆ A query is an expression
  - ◆ Expressions are evaluated w.r.t. a context
  - ◆ The navigation is described using Location Paths
  - ◆ A Location Path consists of:
    - a context node
    - a series of Location Steps separated by /
  - ◆ A Location Step consists of:
    - an axis, a node test, a list of predicates
- document("file.xml")/child::book[attribute::ISBN=10]/  
descendant::section/[position()=1]

D.Florescu, J. Siméon

Int. Conf. On Very Large Databases, Cairo, 2000 96

## XPath in action

- ◆ Location step:
  - an axis, a node test, a list of predicates
- ◆ 13 Axes:
  - ancestor, ancestor-or-self, attribute, child, descendant, descendant-or-self, following, following-sibling, namespace, parent, preceding, preceding-sibling, self
- ◆ Node Test:
  - name test (e.g. section, \*) or type test (e.g. text() )

document("file.xml")/child::\*[attribute::ISBN=10]/  
descendant::section[position()=1]/child::text()

D.Florescu, J. Siméon

Int. Conf. On Very Large Databases, Cairo, 2000 97

## XPath abbreviated syntax

book	CN/child::book
//section/*	CN/descendent::section/child::*
book/@ISBN	CN/child::book/attribute::ISBN
section[1]	CN/child::section[position()=1]
.	CN
../text()	CN/parent::*/child::text()
//section	CN/descendent-or-self::section
//	ROOT/descendent-or-self::*
//section[last()]	CN/descendent-or-self::section[position()=5]
//section[5][title="introduction"]	
//section[title="introduction"][5]	