# Introduction to XML

Service Oriented Architectures

Module 1 - Basic technologies

Unit 1 – Introduction

**Ernesto Damiani**

Università di Milano

# Outline

1. Why XML evolved
2. What is XML?
3. XML 1.0 fundamentals
4. XML family of languages
5. How to use XML
6. Naming conventions: Namespaces
7. Hypertext facilities: XPath, XPointer, XLink
8. Querying and transforming: XQuery and XSLT
9. Applications
10. Challenges

## 1. Why XML evolved

1960-1980  Infrastructure for the Internet

1986  SGML (Standard Generalized Markup Language) for defining and representing structured documents

1991  WWW and HTML introduced for the Internet

1991  Business adopts the WWW technology; huge expansion in the use of the Internet

1995  New kinds of businesses evolve, based on the connectivity of people all over the world and connectivity of applications built by various software providers (B2C, B2B)

Urgent need for a new, common data format for the Internet

# 1. Why XML evolved

- **Needs:**
  - Simple, common rules that are easy to understand by people with different backgrounds (like HTML)
  - Capability to describe Internet resources and their relationships (like HTML)
  - Capability to define information structures for different kinds of business sectors (*unlike* HTML, like SGML)

## Needs (cont’d):

- Format formal enough for computers and clear enough to be human-legible (like SGML)
- Rules simple enough to allow easy building of software (*unlike* SGML)
- Strong support for diverse natural languages (*unlike* SGML)

# 2. What is XML?

## XML = Extensible Markup Language

A set of rules for defining and representing information as structured documents for applications on the Internet; a restricted form of SGML

T. Bray, J. Paoli, and C. M. Sperberg-McQueen (Eds.), Extensible Markup Language (XML) 1.0,
W3C Recommendation 10- February-1998,
http://www.w3.org/TR/1998/REC-xml-19980210/.

T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler (Eds.), Extensible Markup Language (XML) 1.0 (Second Edition),
W3C Recommendation 6 October 2000,
http://www.w3.org/TR/2000/REC-xml-20001006/.

# 2. What is XML?

- **Rule 1: Information is represented in units called *XML documents.***

- **Rule 2: An XML document contains one or more *elements.***

- **Rule 3: An element has a name, it is denoted in the document by explicit markup, it can contain other elements, and it can be associated with *attributes.***

and lots of other rules ...

# Example of an XML document

```
<?xml version="1.0"?>
<catalog>
    <product category = "mobile phone">
        <mfg>Nokia</mfg><model>8890</model>
        <description>
                Intended for EGSM 900 and GSM 1900  networks …
        </description>
        <clock setting= "nist" alarm = "yes"/>
    </product>
    <product category = "mobile phone">
        <mfg>Ericsson</mfg><model>A3618</model>
        <description>...</description>
     </product>
     ...
</catalog>
```

## 2. What is XML?

# Classes of text markup

*descriptive* (the previous example)

*presentational*

**Mobile phones:**

*Nokia 8890*

*Ericsson A3618*

*procedural*

```
<document>
  <newPage style="box"/>
  <bold>Mobile phones:</bold>
  <list>
     <newItem/><italic>Nokia   8890</italic>
     <newItem/><italic>Ericsson  A3618</italic>
  </list>
</document>
```

XML is primarily for descriptive markup.

# 2. What is XML?

**XML is a metalanguage, not a specific language.**

**Defines the rules how to mark up a document — does not define the names used in markup**

**Includes capability to prescribe a *Document Type Definition* (*DTD*) to constrain the markup permitted in a class of documents**

**Intended for *all* natural languages, regardless of character set, orientation of script, etc.**

## Physical (storage) units

**XML document comprises one or more *entities***

- A “document entity” serves as the root
- Other entities may include
  - –external portion of DTD
  - –parsed character data, which replaces any references to the entity
  - –unparsed data

**Entities located by URIs**

# 3. XML 1.0 fundamentals

**XML declaration:** `<?xml version=”1.0”?>`

## Logical data components:

- Markup vocabulary: elements, attributes

```
<product  category = "mobile phone">
            <mfg>Nokia</mfg><model>8890</model>
            ... <clock  setting = "nitz" alarm = "yes"/> ...
</product>
```

- White space
- Parsed and unparsed character data
- Entity references &diagram;
- Comments <!-- how interesting... -->
- Processing instructions

```
<?xml-stylesheet  href="catalog-style.css"  type="text/css"?>
```

# 3. XML 1.0 fundamentals

- **Markup declarations (the DTD):**
  - Internal *vs*. external declarations
  - Root document type
  - Element types: EMPTY, children, mixed, ANY

    ```
    <!ELEMENT category      (mfg, model, description , clock?)>
    <!ELEMENT description  (#PCDATA | feature)*>
    <!ELEMENT clock            EMPTY>
    ```

  - Attribute types: CDATA, ID, IDREF(-S),
    ENTITY(-TIES), NMTOKEN(-S)

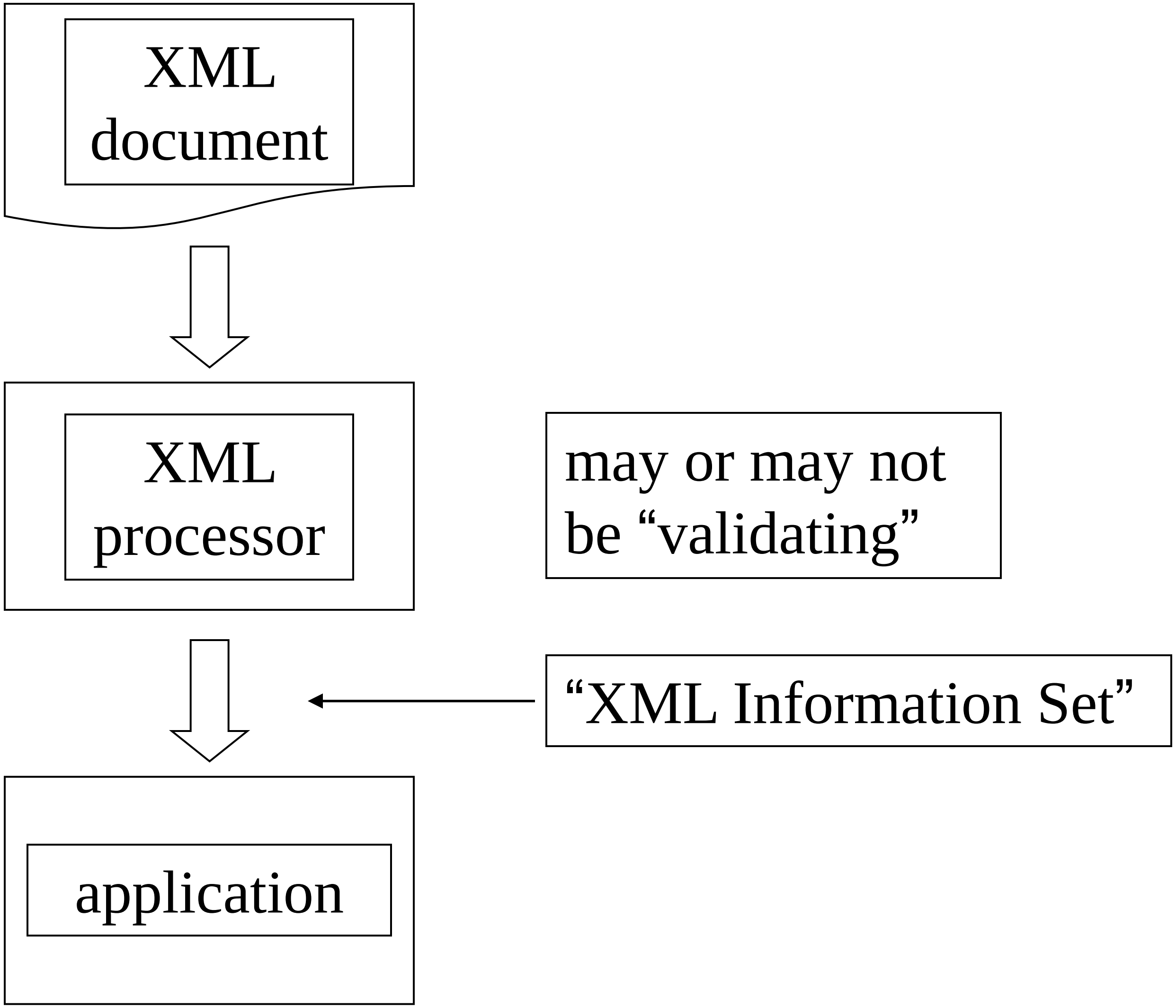
    ```
    <!ATTLIST clock  setting CDATA          #IMPLIED
                     alarm  (yes, no, dual)  "yes"    >
    ```

  - Notations
  - Entities

# 3. XML 1.0 fundamentals

- **Conformance:**
  - *Well-formed:*
    - –syntactically correct tags
    - –matching tags
    - –nested elements
    - –all entities declared before they are used
  - *Valid*:
    - –well-formed
    - –DTD + doctype matches DTD
    - –unique IDs
    - –no dangling IDREFs

# 3. XML 1.0 fundamentals



XML document

XML processor

may or may not be “validating”

“XML Information Set”

application

# 3. XML 1.0 fundamentals

- Example:

  **eXtensible Customer Information Language (xCIL)**

  - OASIS Customer Information Quality Committee

> "… reliable and accurate customer information is now more than ever essential in establishing effective customer relationships ... need to develop a standard way of describing Customers (e.g., Identity, Name, Address, etc.)."

  - Built on
    - eXtensible Name and Address Language (XNAL)
    - two sub DTDs
      - eXtensible Name Language (xNL)
      - eXtensible Address Language (xAL)

- **See details in next section of notes**

# 4. XML family of languages

- **Specification of XML 1.0 was just the first step in the development of languages for the management of data on the Web.**

**http://www.w3.org/TR/ (W3C Technical Publications)**

- **XML-related languages fall into the following classes:**
  - XML accessories
  - XML transducers
  - XML applications

(See third sectin of notes for details)

# XML Accessory

- Extends the capabilities specified in XML
- Intended for wide, general use

Examples:

- XML Names: to allow qualified names
- XML Schema: extends the definition capabilities
- XPath: for addressing parts in XML documents
- XLink: to create hyperlinks between resources

# XML Transducer

- Converts XML input data into output
- Associated with a processing model

Examples:

- XSLT: for document transformations
- CSS: for rendering
- XQuery: for querying

# XML Application

- Defines constraints for a class of XML data
- Intended for a specific application area

Examples (developed at W3C):

- XHTML: reformulation of HTML 4.0
- RDF: to describe metadata for resources
- XML-Signature: for digital signatures
- XForms: for Web forms

# XML in an organization

- **Various ways XML might occur:**
  - Encoding format for uninterpreted data
  - Format for data interchange
  - Format for managing information assets

# 5. How to use XML

## Uninterpreted data

- Data accessed from the Internet and forwarded with little or no processing to another application

# 5. How to use XML

- **Data interchange**
  - Encoding of data
    - on import, useful for dissecting data
    - for export, provides added value to users
  - Encoding of protocols
    - designating functions and marshalling arguments

  For diverse applications
    - within an organization, to support integration
    - among business partners, to support business collaboration; requires negotiated agreement between partners

# Information assets

- Maintained in an XML document repository
- Requires either development of proprietary procedures for accessing and maintaining assets or adoption (and adaptation) of rules developed elsewhere
- May require major changes in organization’s work processes

# 6. Naming conventions

- **Names play a key role**
  - For elements, attributes, entities and notations
  - May be introduced (and limited) by a DTD
- **Often need to use elements and attributes originating from different environments (or applications)**
  - Vocabularies in two environments may include common names intended for different purposes
  - If multiple declarations used in a single DTD, name collisions must avoided

# 6. Naming conventions

## Example

- **From BiblioML**

```
<!-- A title of work or in a related title element. -->

<!ELEMENT Title  (#PCDATA | NonSortingData)*   >
<!ATTLIST Title
  Type  (Proper | AnotherAuthor | Parallel | OtherInfo) #IMPLIED
  ISOLanguage      CDATA                                #IMPLIED
  Language         CDATA                                #IMPLIED >
```

- **From xNL**

```
<!ELEMENT Title  (#PCDATA)   >

<!-- attribute defines type of title, eg. Sex, Honorary, Profession, etc. --
 >
<!ATTLIST Title
  Type             CDATA                                #IMPLIED>
```

6. Naming conventions

## XML namespaces

- **Provides a method for qualifying element and attribute names so that name collisions can be avoided**
- **Motivation: modularity and documentation**

> If a well-understood markup vocabulary for element and attribute names exists, it should be re-used rather than re-invented, *especially* if there is also software available.

# XML namespaces

- **Collection of names, identified by a URI**
  - No formal rules for defining names in a namespace

Example

- *Namespace*: http://uwaterloo.ca
- *Element names*: department, name, professor, student, last_name, first_name, ...
- *Global attribute names*: id, ...
- *Per-element-type attribute names*:  student: supervisor, ...

# 6. Naming conventions

- **Namespace declaration: defines a label (prefix) for the namespace and associates it to the namespace identifier (URI)**
- **Qualified name: a namespace prefix and a local part, separated by a colon**

```
<?xml version="1.0"?>
<report xmlns:uw="http://uwaterloo.ca">
   <uw:department>
      <uw:name>Department of Computer Science</uw:name>
       ...
</report>
```

# 7. Hypertext facilities

**Hyperlink: cross-reference primarily for presentation to a human**

- Arc: two ends + direction ⇒ *source* and *destination*
- Traversal: following a link (for any purpose)

**HTML (review):**

- "HyperText Markup Language" for WWW
- Supports simple (binary, unidirectional) links
  - special elements <A> can be used as anchors
  - named anchor (identified by *id* or *name* attribute) used as destination
  - anchor with *href* attribute is link source
  - destination name matches href value (URI)

# 7. Hypertext facilities

## Examples

For more information about links in HTML, please consult the <a href="http://www.w3.org/TR/html401/struct/links.html"> HTML 4.01 specification</a>

<h2><a name="h-12.1">12.1</a> <a name="links">Introduction to links</a> and <a name="anchors">anchors</a></h2>

<a href="http://www.w3.org/TR/html401/struct/links.html#links"> Introduction to links and anchors</a>

<div class="navbar"> <a href="tables.html">previous</a> <a href="objects.html">next</a> <a href="../cover.html#minitoc"> contents</a> <a href="../index/elements.html">elements</a> <a href="../index/attributes.html">attributes</a> <a href= "../index/list.html">index</a> </div>

# 7. Hypertext facilities

```
<a
href="http://www.w3.org/TR/html401/struct/links.html#links">Introduction
to links and anchors</a>
```

- **An HTML link has the following characteristics:**
  - Comprises only one arc
  - Expressed at source end
  - Identifies destination end via URI
    - N.B.: server has freedom in finding or dynamically creating destination
  - Users can initiate traversal only from source end
  - Hyperlink's effect (on windows, frames, go-back lists, stylesheets in use, and so on) determined by user agents, not by hyperlink itself

# 7. Hypertext facilities

- **XML not designed specifically for hypertext**
- **Nevertheless, XML provides:**
  - references to external entities (by URI references)

```
<!ENTITY spring   SYSTEM "../grafix/flower.gif"  NDATA gif >...
 <figure picture="spring">
```

  - references to elements in the same document (via ID and IDREF attributes)

```
<!ELEMENT student (name address*)>
<!ATTLIST   student  id              ID         #REQUIRED
                     sibling-of   IDREF   #IMPLIED      >
<student id="123"><name>Mikko</name></student>
<student id="425" sibling-
 of="123"><name>Lisa</name></student>
```

# 7. Hypertext facilities

- **Advanced techniques for referencing & linking:**
  - **XPath**
    - –for addressing parts of XML document, typically **node-sets**
    - –intended to be used by other specifications
  - **XPointer**
    - –extends addressing capabilities of XPath to include **locations** *within* unstructured (i.e., text) components
    - –locations are points or ranges
  - **XLink**
    - –for specifying links between resources
    - –made explicit by an XLink linking element
    - –link ends described by XPointer

## XPath

- Single step: “axis::node-test[predicate]”
  - axes: child, descendant, self, ancestor, following-sibling, etc.
- Path: sequence of steps separated by “/”
- Abbreviated notation

```
descendant-or-self::node( ) /
  child::book[attribute::year=2000] / child::price
//book[@year=2000]/price
```

# 7. Hypertext facilities

**XLink's simple link**

- One arc only
- From (implicit) local resource to remote resource
- May also include metadata

**Example**:

```
<icr:attendees
        xmlns:icr="http://icr.uwaterloo.ca/"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xlink:type="simple"
        xlink:href="students.xml"
        xlink:role="http://icr.uwaterloo.ca/courses/attendees"
        xlink:title="Attendee List for XML Short Course"
        xlink:show="new"
        xlink:actuate="onRequest">
Participants from Industry
    </ icr:attendees >
```

# 7. Hypertext facilities

**XLink's extended link**

- Associates an arbitrary number of resources
- Any combination of remote and local resources
- Separate elements specify resources and arcs
  - <… xlink:type="locator" xlink:label="..." …> for remote resources
  - <… xlink:type="resource" xlink:label="..." …> for local resources
  - <… xlink:type="arc" xlink:from="..." xlink:to="..." …> for arcs

**Often stored separately from the resources they associate ⇒ linkbases**

# 7. Hypertext facilities

## Example

```
        <icr:class
    xmlns:icr="http://icr.uwaterloo.ca/"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xlink:type="extended"
    <icr:attendees            xlink:type="locator"  xlink:label="class"
                              xlink:href="students.xml"
                              xlink:role="http://icr.uwaterloo.ca/courses/attendees"
                              xlink:title="Attendee List for XML Short Course" />
    <icr:instructors          xlink:type="resource"  xlink:label="teacher"
                              xlink:title="Attendee List for XML Short Course" >
       Joe Smith </icr:instructors>
    <icr:enrol    xlink:type="arc"  xlink:from="teacher"  xlink:to="class"
                              xlink:show="replace"  xlink:actuate="onRequest" />
    Participants in XML Short Course
  </ icr:class >
```

# 8. Querying and transforming

- **Roots in conventional database querying + information retrieval + Web search engines**
- **David Maier's desiderata**
  - Preserve order and association; produce XML output
  - Suitable for documents, business records, and metadata
  - Support for new datatypes
  - Support {selection, extraction, reduction, restructuring, combination}
  - No schema required, but exploit available schema
  - XML representation; mutually embedding with XML; programmatic manipulation
  - XLink and XPointer cognizant
  - Namespace alias independence

## XML Query Specifications

- ***XML Query Requirements*, 16 February 2001**
- ***XML Query Use Cases* , 8 June 2001**
- ***XQuery 1.0 and XPath 2.0 Data Model* (replaces XML Query Data Model), 7 June 2001**
- ***XQuery 1.0 Formal Semantics* (replaces XML Query Algebra), 7 June 2001**
- ***XQuery 1.0: An XML Query Language*, 7 June 2001**
- ***XML Syntax for XQuery 1.0* (*XQueryX*), 7 June 2001**

## Use Case 1.1.9.10 Q10

- In the document “prices.xml”, find the minimum price for each book, in the form of a “minprice” element with the book title as its title attribute.
- Solution in XQuery:

```
            <results>
      {   LET $doc := document("prices.xml")
          FOR $t IN distinct($doc/book/title)
          LET $p := $doc/book[title = $t]/price
          RETURN
            <minprice title={ $t/text() }>
                  {  min($p)  }
            </minprice>
      }
    </results>
```

# 8. Querying and transforming

## XSLT

### XML Stylesheet Language Transformations

- For transforming XML documents, including converting XML to XHTML, etc.

  Example: For each employee, process the name and each group in the employee’s department.

```
          <xsl:template match="employee">
            <fo:block>
              Employee <xsl:apply-templates select="name"/>
belongs to group
              <xsl:apply-templates
select="ancestor::department/group"/>
            </fo:block>
          </xsl:template>
```

# 9. Applications

- **Many, diverse XML applications**
- **Typically resulting from industry initiatives**

- **Examples:**
  - eCatalog XML (eCX): a DTD for catalogs, developed to address the problem of exchanging product information between different catalog systems; version 2.0 published May 2000.
  - Open Catalog Protocol and Format (OCP/OCF) a software protocol to exchange complex data between product catalogs and a DTD for a catalog.

# 9. Applications

- BASDA eBIS-XML: for exchange of standard business documents; enables the direct exchange of purchase orders and invoices and other business documents between different software packages, via e-mail and the Internet, without the need for EDI (Electronic Data Interchange) middleware
- ebXML: to enable a global electronic marketplace “where enterprises of any size and in any geographical location can meet and conduct business with each other through the exchange of XML based messages”

# 9. Applications

- Electronic Commerce Modeling Language (ECML): a set of hierarchical payment oriented structures “to enable automated software, including electronic wallets, from multiple vendors to supply needed data in a more uniform manner”; ECML v2 DTD published in Feb. 2001
- VISA XML Invoice Specification: a DTD for VISA invoices, “intended to increase a corporation's ability to automate B2B purchasing functions and monitor travel and entertainment expenses worldwide”

# 9. Applications

- CallXML: a phone markup language, used to describe the user interface of a telephone, voice over IP, or multi-media call application to a CallXML browser so it can control and react to the call itself.
- Theological Markup Language (ThML)
- Chemical Markup Language (CML)
- Mathematical Markup Language (MathML)
- etc.

# 9. Applications

- Human Markup Language (HumanML): "To promote XML standards that will reduce human misunderstanding in society... through the explicit markup of various communication constructs including thought, emotion, purpose, and motivation"

***Many* more listed at**

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

# 10. Challenges

- **Knowledge of XML or a specific XML application *not* sufficient for using XML technology; usually expertise in other members of the XML family also needed**

- **Continuous changes in specifications and in software; many important specifications still working drafts**

- **Parallel development of related /competing specifications at W3C and industry sectors**

## 10. Challenges

- **Requires effective collaboration among business partners, either within a single industry sector or among partners crossing industry boundaries; good communication skills needed**
- **Partners may need to use (and depend on) XML specifications before W3C or industry sector(s) has finalized them.**