Introduction to XML

Jaana Holvikivi
Metropolia

Content

- Defining XML
- XML structure
- Application areas
- Schema example
- XML rules: well-formed XML

XML = Extensible Markup Language

- General mark-up language, a metalanguage
- forms a family of standards
- based on SGML
- has many uses and possibilities when combined with other standards, languages and products
- W3C recommendation
  - version 1.0
  - 6.10.2000
  - a set of rules to combine, exchange and publish information

XML – metalanguage

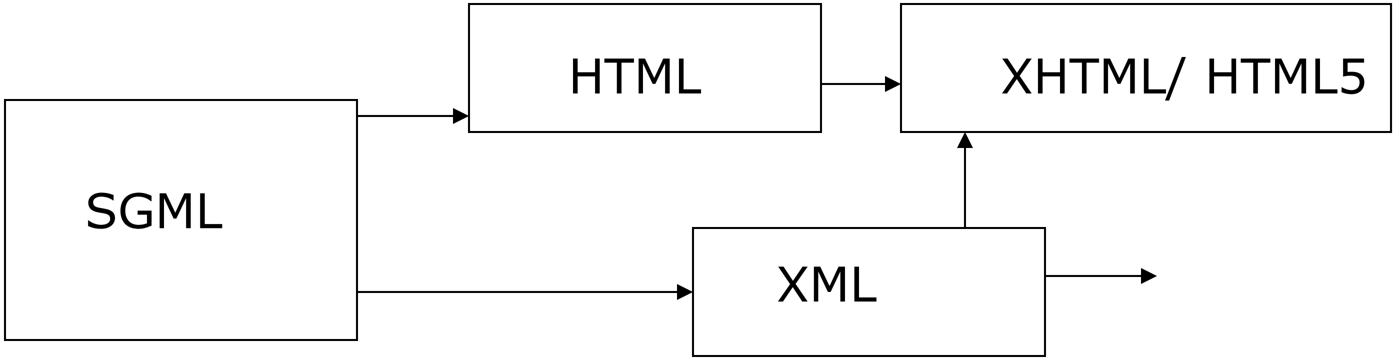
- the universal format for structured documents and data on the Web
- XML makes it easy for a computer to generate data, read data, and ensure that the data structure is unambiguous
- readable for both human and computer:
- text format: it allows people to look at the data without the program that produced it: no parser needed
- platform and programming language independent

W3C World Wide Web Consortium

- created in October 1994 to lead the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability
- about 400 Member organizations
- ”The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential as a forum for information, commerce, communication, and collective understanding.”
- has developed more than 40 technical specifications (like HTML)
- open source software

XML - SGML - HTML

- XML combined features from SGML and HTML
- many tools
- XHTML and HTML5 follow XML recommendation
- cannot solve all problems alone
- all three languages are needed (XML, HTML, SGML)



XML –document instance

```
<?xml version="1.0"?>
<!-- Example of an document instance  -->
<university>
  <department>
       <name>
              Department of Genetic Engineering
       </name>
       <address>
              DNA St 2
       </address>
  </department>
</university>
```

XML is for structuring data:

- Structured data : spreadsheets, data transfer, configuration parameters, financial transactions, technical drawings, data base, etc.
- XML is a set of rules (you may also think of them as guidelines or conventions) for designing text formats that let you structure your data.
- XML is not a programming language
- extensible, platform-independent, and supports internationalization and localization: XML is fully Unicode-compliant

XML looks a bit like HTML

- tags / elements and attributes
- XML uses the tags only to delimit pieces of data, and leaves the interpretation of the data completely to the application that reads it
- <p>

XML files are text files that people shouldn't have to read

- the rules for XML files are strict,
- The official XML specification forbids applications from trying to second-guess the creator of a broken XML file
- XML is verbose by design (compared with JSON)

Example: android

```
<level-list
xmlns:android="http://schemas.android.com/apk/res/android">

  <item android:maxLevel="0"
android:drawable="@drawable/ic_wifi_signal_1" />
  <item android:maxLevel="1"
android:drawable="@drawable/ic_wifi_signal_2" />
  <item android:maxLevel="2"
android:drawable="@drawable/ic_wifi_signal_3" />
  <item android:maxLevel="3"
android:drawable="@drawable/ic_wifi_signal_4" />
 </level-list>
```

Example of XMPP Client-side:

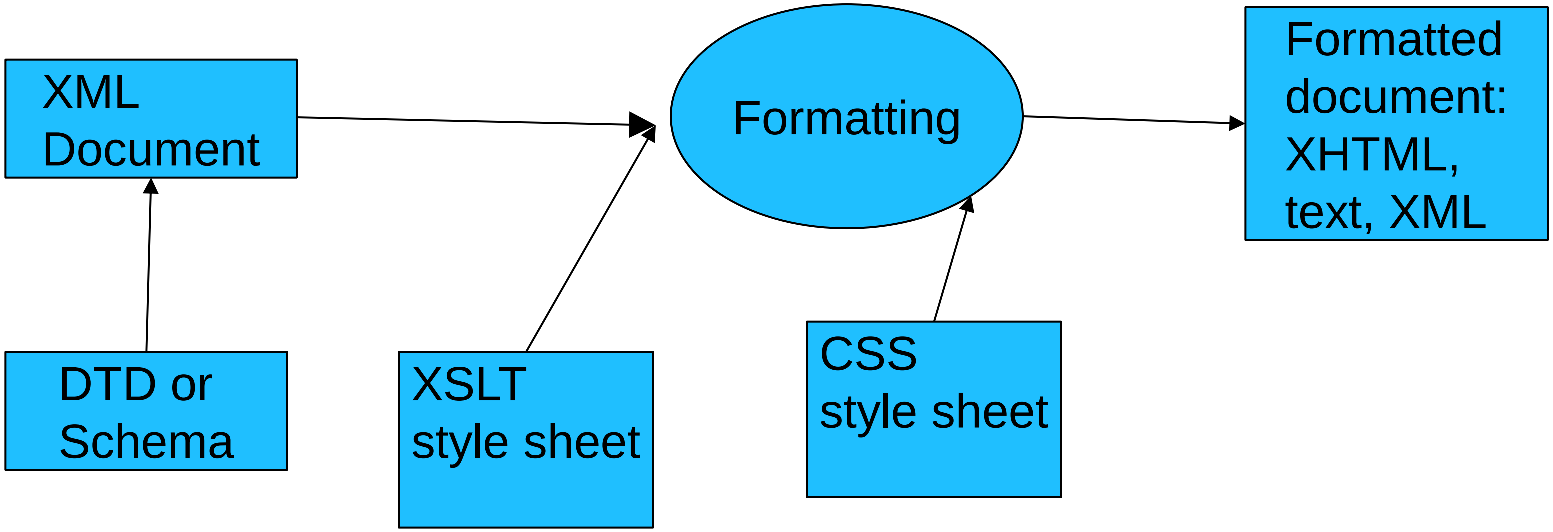
```
<starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>

<stream:stream
    from='juliet@im.example.com'
    to='im.example.com'
    version='1.0'
    xml:lang='en'
    xmlns='jabber:client'
    xmlns:stream='http://etherx.jabber.org/streams'>

<message from='juliet@im.example.com'
         id='ju2ba41c'
         to='romeo@example.net'
         type='chat'
         xml:lang='en'>
    <body>Art thou not Romeo, and a Montague?</body>
  </message>

</stream:stream>
```

XML is a family of technologies

- XML 1.0
- Schemas: to define the rules
- Namespaces to refer to schemas
- Xpath – language for navigation
- XSL and XSLT transformations
- DOM interfaces
- DTDs and CSS are used together with XML standards

- XML is license-free, platform-independent and well-supported!!

Using XML standards



XML schema for XMPP client

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
   xmlns:xs='http://www.w3.org/2001/XMLSchema'
   targetNamespace='jabber:client’    xmlns='jabber:client'
   elementFormDefault='qualified'>

  <xs:import  namespace='urn:ietf:params:xml:ns:xmpp-stanzas'/>

  <xs:element name='message'>
    <xs:complexType>
      <xs:sequence>
        <xs:choice minOccurs='0' maxOccurs='unbounded'>
          <xs:element ref='subject'/>
          <xs:element ref='body'/>
          <xs:element ref='thread'/>
        </xs:choice>
        <xs:any     namespace='##other'
                    minOccurs='0'
                    maxOccurs='unbounded'
                    processContents='lax'/>
        <xs:element ref='error'
                    minOccurs='0'/>
      </xs:sequence>
```

… continues

XML schema for XMPP client

Cont:

```
<xs:attribute name='from'
           type='xs:string'
           use='optional'/>
    <xs:attribute name='id'
           type='xs:NMTOKEN'
           use='optional'/>
    <xs:attribute name='to'
           type='xs:string'
           use='optional'/>
    <xs:attribute name='type'
           use='optional'
           default='normal'>
     <xs:simpleType>
      <xs:restriction base='xs:NMTOKEN'>
       <xs:enumeration value='chat'/>
       <xs:enumeration value='error'/>
       <xs:enumeration value='groupchat'/>
       <xs:enumeration value='headline'/>
       <xs:enumeration value='normal'/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute ref='xml:lang' use='optional'/>
   </xs:complexType>
 </xs:element>
```

continues

XML application areas

- Data transfer
  - mobile apps (android), RSS
  - relational databases
  - EDI (Electronic Data Interchange)
- E-commerce: B2B and B2C, Web Services
- Publishing
  - Electronic documents
  - Metadata, DocBook
- Semantic web
- Internal format in browsers (HTML)
- Microsoft: .NET and internal format for Office
- GIS
- Ajax and XMLHTTP, Googlemaps

Document instance

- contains the information of the document, marked-up according to agreed rules
- self-descriptive tags
- helps in interpretation of data
- elements and child elements
- text and comments

XML markup

Document instance

- elements and child elements
- attributes
- entities
- processing instructions
- text and comments

Elements

- Part of logical document structure
- delimiting tags
  - opening tag
  - closing tag
- element name
- element contents
  - child elements or text
- examples:

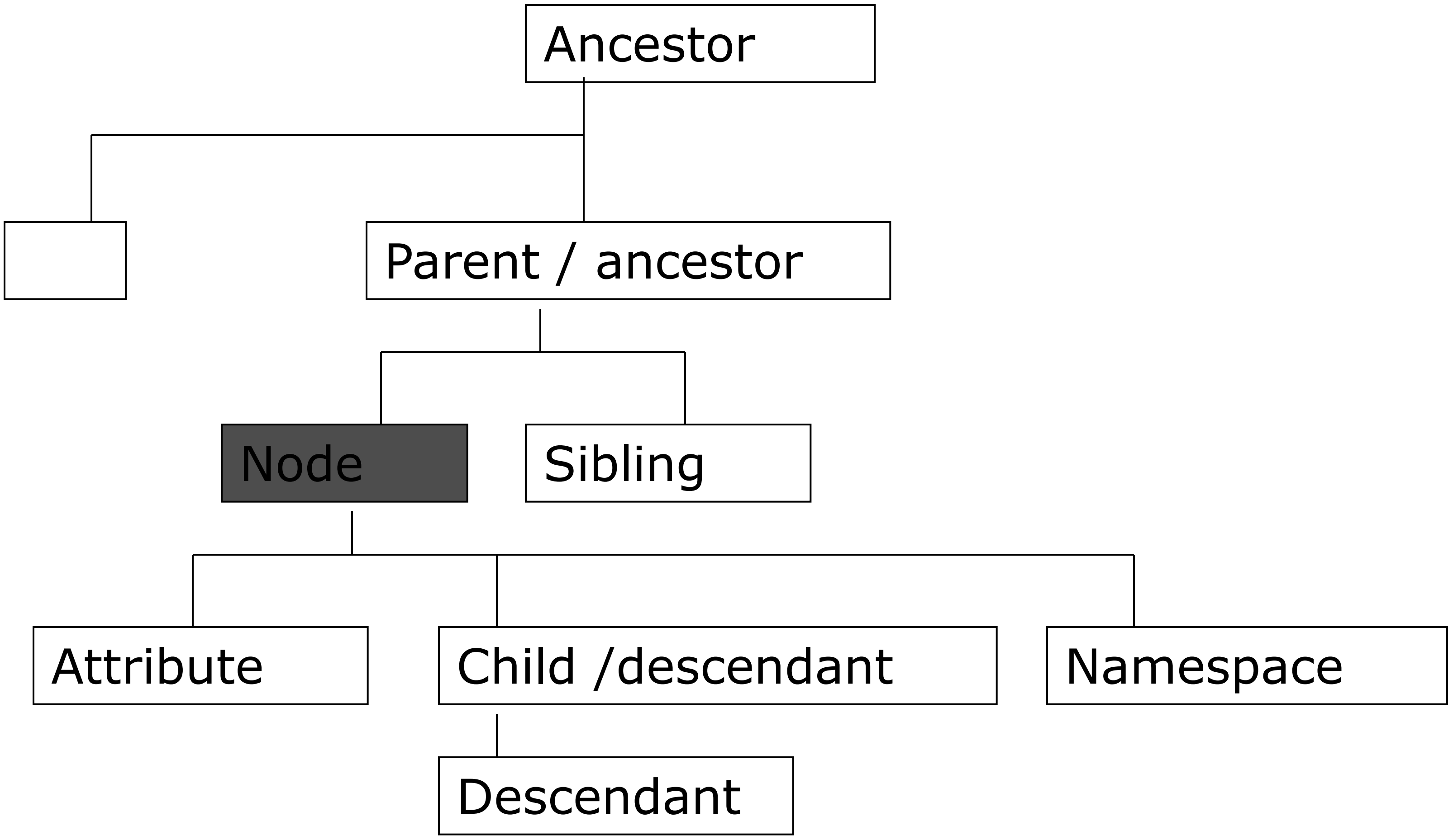
```
<capital>Helsinki</capital>

<country>
  <cname>Finland</cname>
  <capital>Helsinki</capital>
</country>
```

Element nesting: rules

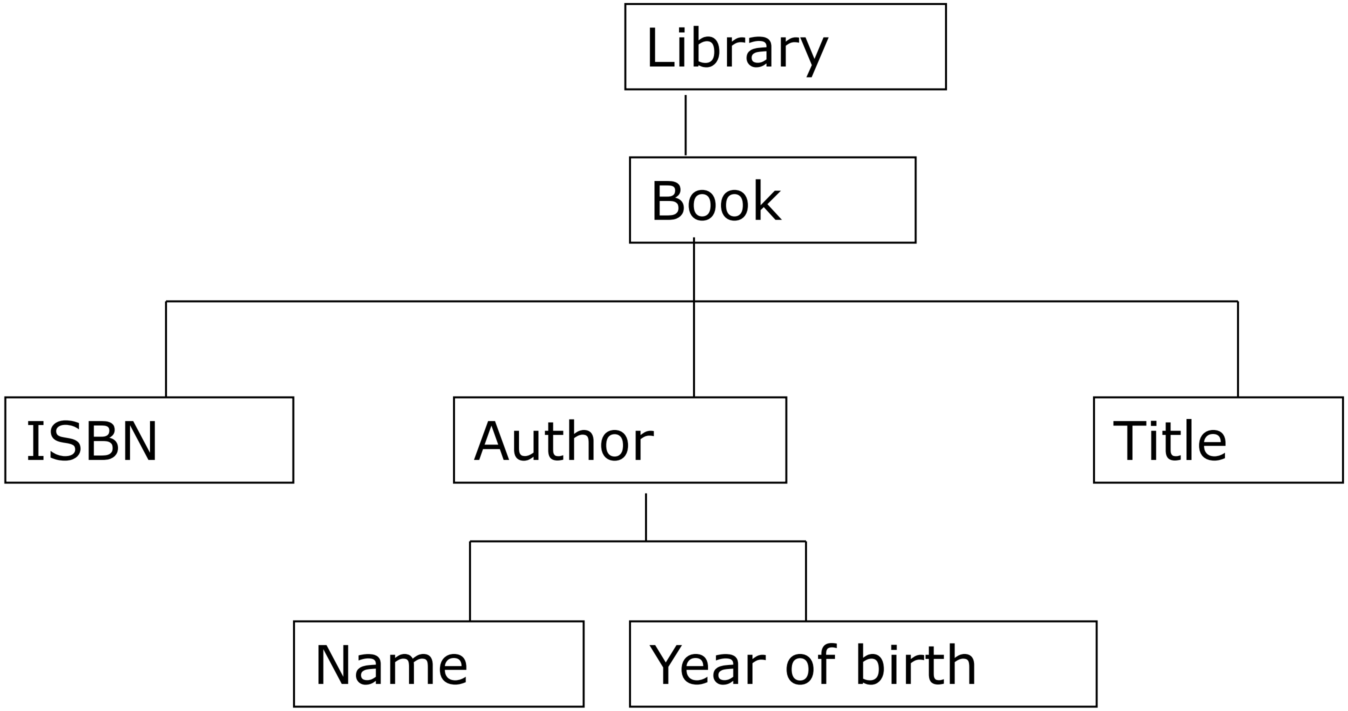
- opening tag and closing tag must match
- element must be completely within another (no crossed tags)
- element hierarchy
  - root = document element - only one!
  - Tree structure

- case-sensitive: capitals are different characters than lower case letters
  - <chapter> not same as <Chapter>
- element names must follow XML rules
- **= well-formed**

Document tree



Document tree



XML elements must follow these naming rules

- Names can contain letters, numbers, and other characters
- must not start with a number or punctuation character
- must not start with the letters xml (or XML or Xml ..)
- cannot contain spaces or colons
- Follow these simple rules:
  - Any name can be used, no words are reserved, but the idea is to make names descriptive.
  - Examples: <first_name>, <last_name >.
- Which of the following are valid?
  - <first.name> <xml-root> <123> <Big Apple>
  - <p>paragraph</P>

Element contents

- An element can have
  - element content,
  - mixed content,
  - simple content, or
- empty content
  - <nothing></nothing>
  - <useless/>
- why
  - content could be elsewhere
  - the empty element has a reference

```
<image file="pict.jpg"/>
```

Attributes

- Element property or contents
- attached to opening tags (or empty element tags)
  - attribute name
  - attribute value
- only one value
- the value can contain any characters
- are they needed ?

```
<book author=”Oscar Wilde">
...
</book>

<book keywords="XML SGML">
...
</book>
```

Processing instructions

- Processing instruction is an instruction within the XML document (which is not part of the actual document but which is passed up to the application)
- delimiters **<?** and **?>**
- example (almost): XML declaration:

```
<?xml version="1.0" encoding="UTF-8"
    standalone="yes"?>
```

  - version is 1.0
  - character set
  - no external definitions

Mark-up declarations

- Commands to the XML processor
  - start: <!
  - End: >
  - document type structure, document parts, etc.

    <!DOCTYPE pizzas SYSTEM "pizzas.dtd">
- Comments
  - <!-- This is a comment -->

XML -processors

- A software module called an XML processor is used to read XML documents and provide access to their content and structure
- XML parser
  - finds errors
  - produces information for other applications
- an XML processor is doing its work on behalf of another module, called the application