University of Alicante

Network Europe - Russia - Asia of Masters in Informatics as a Second Competence
159025-TEMPUS-1-2009-1-FR-TEMPUS-JPCR
Master

- University Master's Degree In Applications Development And Web Services
- Aim:
  Teaching students how to develop web-based computer systems and large business systems, applying best practices to software engineering.
- Professions for which the degree qualifies its holder:
  - Analyst, software architect, project manager, software designer, programmer.

Master

- The course also contains two optional routes, each focusing on one of the two following aspects:
  - Analysis and Development of New Technologies for the Internet
  - Developing Large Systems
Master

- **Analysing and Developing New Technologies for the Internet** focuses on technologies that substantially improve interaction and usability of traditional web-based application user interfaces.

Master

- **Developing Large Systems** focuses on learning two types of applications – enterprise resource planning systems and developing large mainframe systems.
MASTER'S DEGREE COURSE - CREDITS AND SUBJECTS

<table>
<thead>
<tr>
<th>Type of subject</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compulsory (OB)</td>
<td>41</td>
</tr>
<tr>
<td>Optional (OP)</td>
<td>11</td>
</tr>
<tr>
<td>Final project (OB)</td>
<td>8</td>
</tr>
<tr>
<td><strong>TOTAL CREDITS</strong></td>
<td><strong>60</strong></td>
</tr>
</tbody>
</table>

Distribution of subjects by module

<table>
<thead>
<tr>
<th>ADVANCED SOFTWARE ENGINEERING</th>
<th>ADVANCED SOFTWARE DEVELOPMENT METHODOLOGIES</th>
<th>08</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADVANCED SOFTWARE SYSTEMS ANALYSIS</td>
<td></td>
<td>08</td>
<td>3</td>
</tr>
<tr>
<td>PATTERN-LED DESIGN</td>
<td></td>
<td>08</td>
<td>3</td>
</tr>
<tr>
<td>USER INTERFACE DESIGN</td>
<td></td>
<td>08</td>
<td>2</td>
</tr>
<tr>
<td>QUALITY TESTING AND CONTROL</td>
<td></td>
<td>08</td>
<td>3</td>
</tr>
<tr>
<td>SOFTWARE SYSTEMS ARCHITECTURE</td>
<td></td>
<td>08</td>
<td>3</td>
</tr>
</tbody>
</table>
### Distribution of subjects by module

#### SOFTWARE SYSTEM TECHNOLOGIES

<table>
<thead>
<tr>
<th>Subject</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Programming in Desktop Environments</td>
<td>4</td>
</tr>
<tr>
<td>Developing Web Applications</td>
<td>3</td>
</tr>
<tr>
<td>Developing Distributed Applications</td>
<td>3</td>
</tr>
<tr>
<td>Programming Mobile Devices</td>
<td>2</td>
</tr>
<tr>
<td>XML</td>
<td>2</td>
</tr>
<tr>
<td>Databases</td>
<td>3</td>
</tr>
</tbody>
</table>

#### MASTER'S WEB

### Distribution of subjects by module

#### SERVER ADMINISTRATION

<table>
<thead>
<tr>
<th>Subject</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Servers</td>
<td>2</td>
</tr>
<tr>
<td>Application Servers</td>
<td>2</td>
</tr>
</tbody>
</table>

#### DEVELOPING LARGE SYSTEMS

<table>
<thead>
<tr>
<th>Subject</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developing Systems in Mainframe Environments</td>
<td>7</td>
</tr>
<tr>
<td>Developing Management Systems for Business Resources</td>
<td>4</td>
</tr>
</tbody>
</table>

#### ANALYSIS AND DEVELOPMENT OF NEW TECHNOLOGIES FOR THE INTERNET

<table>
<thead>
<tr>
<th>Subject</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysing Internet Trends</td>
<td>3</td>
</tr>
<tr>
<td>Browser Orientated Internet Technologies</td>
<td>3</td>
</tr>
<tr>
<td>Developing Rich Internet Interfaces</td>
<td>5</td>
</tr>
</tbody>
</table>

#### MASTER'S FINAL PROJECT

<table>
<thead>
<tr>
<th>Subject</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed Web Application Project</td>
<td>8</td>
</tr>
</tbody>
</table>
Master

- Credits: 60 (1 year)
- Fee: 30,32€/credit (first time)
  - Total: 1.819,20€
What is XML?
And XHTML?
And XPath, XQuery, XSLT, ...?

Has anybody used?
Contents

- Introduction
- XML applications
- Structure of a document
- Other XML technologies
- Software
- XML 1.1

Contents

- Introduction DTD
- Definition of a DTD
- Document Type Declaration
- Validation of a DTD
- Popular DTDs
- Validation of a document
Introduction

- XML
- Versions
- Advantages
- What do I need to use XML
- My first XML document
Wikipedia: Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards.

Wikipedia: The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services.
Extensible Markup Language
World Wide Web Consortium (W3C)
Problems:
- HTML not flexible, mix of content and presentation
- SGML too complex
- Subset of SGML

GML (IBM, 1969)
SGML (ISO 8879, 1986)
XML (W3C, 1998)
Call me at https://www.diarioinformacion.com/ciudadela.html/21 - Noticia Torreblanca

<rss version="2.0">
  <channel>
    <title><![CDATA[Informacion - Alicante]]></title>
    <link>http://www.diarioinformacion.com</link>
    <description><![CDATA[Informacion - Alicante]]></description>
    <copyright><![CDATA[Copyright INFORMACION]]></copyright>
    <lastBuildDate>2009-11-09T06:46:57Z</lastBuildDate>
    <category>Noticias</category>
    <image>
      <title><![CDATA[Informacion]]></title>
      <url>http://www.diarioinformacion.com/favicon.ico</url>
      <link>http://www.diarioinformacion.com</link>
      <description><![CDATA[Informacion - Alicante]]></description>
    </image>
    <item>
      <title><![CDATA[Nick Noce rodriz en Ciudad de la luz]]></title>
      <description><![CDATA[El actor norteamericano participa en el rodaje de 'Mi querido desconocido']]></description>
      <pubDate>2009-11-09T06:46:57Z</pubDate>
    </item>
    <item>
      <title><![CDATA[Un hombre armado toma como rehén a director de una escuela en ENJAI]]></title>
      <description><![CDATA[Las autoridades están negociando con el secuestrador, que podría ser el pai]]></description>
      <pubDate>2009-11-09T06:46:57Z</pubDate>
    </item>
  </channel>
</rss>
Not a language (no predefined tags), XML is a **metalanguage**:

- Defines tags and attributes
- Defines structural relationships

**XHTML**: hybrid HTML + XML

- HTML written according to XML (application of XML)
- Substitute of HTML
XML

Metalanguages

- GML
- SGML

Languages

- HTML
- XHTML
- RSS
- ... ...

Versions

- 10/2/1998: XML 1.0
- 4/2/2004: XML 1.0 Third Edition
- 16/8/2006: XML 1.0 Fourth Edition
- 26/11/2008: XML 1.0 Fifth Edition

- The four editions correct errors and clarify/detail the standard, but they don’t define a new standard
Versions

4/2/2004: XML 1.1
- Updates 1.0 to new Unicode standard
- W3C: “You are encouraged to create or generate XML 1.0 documents if you do not need the new features in XML 1.1; XML Parsers are expected to understand both XML 1.0 and XML 1.1”

- Clarifies/details the standard, but they don’t define a new standard

Versions

Extensible Markup Language (XML) 1.0 (Fifth Edition)
- W3C Recommendation 26 November 2008
- http://www.w3.org/TR/2008/REC-xml-20081126/
- To get the last version:
  - http://www.w3.org/TR/xml/
Advantages

- Provides metadata for data → Improves searches
- Structured data → Allows fine grain updates
- **Separates content (data) / presentation:**
  - Changes of data / presentation are easier
  - Allows multiples views of the same data
La más grave de las enfermedades mentales tiene un nombre equívoco y provisional. Lo que hoy se denomina esquizofrenia probablemente con varias enfermedades con causas y pronósticos distintos. Pero esto no se sabrá con certeza hasta que no se complete más un enrevesado "puzzle" que hay numerosas piezas que tienen que ver con el desarrollo fetal, los factores ambientales y la genética. Ahora se han hecho fotografías en su sitio para esclarecer las causas y mejorar el panorama actual: hoy, como hace milenios, una de cada 100 personas sufre esquizofrenia, y aún hay un 25% que no se recupera ni con tratamiento.

Por estudios en familias de esquizofrénicos se sabe que la enfermedad no obedece a un solo gen. La prueba principal la dan los gemelos genéticamente idénticos (monoizigóticos): sólo un menor de la mitad de los casos son gemelos monoizigóticos, pero tanto los hijos del genotipo no afectado como los del genotipo esquizofrénico tienen un riesgo similar (menor del 20%).
Advantages

- Allows you to create markup languages for specific domains:
  - Chemistry: Chemical Markup Language (CML)
  - Mathematics: Mathematical Markup Language (MathML)
  - Music: MusicXML
  - Monetary information: Open Financial Exchange (OFX)
  - Human resources (job offers, CVs, etc.): HR-XML
Exercise

Find the specification of a markup language based on XML

Advantages

- Self-describing data:
  - Much computer data from the last 40 years is lost because the data format is unknown nowadays
  - XML is a simple data format
  - The removal of bytes does not noticeably corrupt the remaining content
Advantages

- Self-describing data:
  
  ```xml
  <persona id="p110" sexo="m">
  <nombre>Pedro López</nombre>
  <direccion>de los Palotes, 120</direccion>
  <fnacimiento>30/06/1970</fnacimiento>
  </persona>
  
  p110;m;Pedro López;de los Palotes, 120; 30/06/1970
  
  ```

- Improves interchange of data among applications:
  - XML is a non-proprietary format
  - XML is very well documented (no secrets)
  - XML is easy to read and write
  - Example: Open Financial Exchange (OFX)
    - XML application for describing and storing financial data
    - Used by: Quicken, Microsoft Money, GnuCash, etc.
What do I need to use XML

Minimum:
  ◦ To edit:
    • ASCII standard editor:
      • Microsoft Notepad
  ◦ To display:
    • XML compatible browser:
      • Microsoft Internet Explorer 5
      • Netscape 6

Recommended:
  ◦ To edit:
    • Notepad++
    • EditiX
    • XML Spy
  ◦ To display:
    • Microsoft Internet Explorer 8 with IE XML/XSL Viewer Tools
    • Mozilla Firefox 3
My first XML document

- File name:
  - Short and easy to remember
  - Don’t use special characters or space character
    - Better only English alphabet
- File extension: .xml
- Editor:
  - Windows Notepad
- Viewer:
  - Microsoft Internet Explorer or Mozilla Firefox

Example: helloworld.xml

```xml
<?xml version="1.0" ?>
<document>
Hello world!
</document>
```
Exercise

Write an XML document that stores the names of countries and the names of the corresponding capitals.
My first XML document

- countries.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<countries>
  <country name="Kazakhstan" capital="Astana"/>
  <country name="Kyrgyzstan" capital="Bishkek"/>
  <country name="Spain" capital="Madrid"/>
</countries>
```
Error de lectura XML: mal formado
Número de línea 3, columna 36:

<!--country name="Kazakhstan" capital="Astana" -->

Error de lectura XML: etiqueta sin pareja. Se esperaba: </countries>.
Número de línea 6, columna 3:

</countries>
My first XML document

- But there is often more than one way to organize the data, depending on your needs:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<countries>
  <country>
    <name>Kazakhstan</name>
    <capital>Astana</capital>
  </country>
  <country>
    <name>Kyrgyzstan</name>
    <capital>Bishkek</capital>
  </country>
  <country>
    <name>Spain</name>
    <capital>Madrid</capital>
  </country>
</countries>
```
My first XML document

- Attaching a style sheet to an XML document: helloworld-style.xml

```xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="helloworld.css"?>
<document>
Hello world!
</document>
```
My first XML document

- Style sheet: helloworld.css

document {
display: block;
margin: 10%;
font-size: 36pt;
font-weight: bold;
color: blue;
text-align: center;
}
Hello world!

XML applications
XML applications

- Each specific XML-based markup language is called an XML application
  - Application does not mean a program that uses XML
  - Application means the use of XML to a specific domain
- Application = markup language = semantics and vocabulary

Chemical Markup Language:
```xml
<?xml version="1.0" ?>
<cm1 xmlns="http://www.xml-cml.org/schema/cml2/core"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.xml-cml.org/schema/cml2/core/cmlCore.xsd">
  <molecule title="Water">
    <atomArray>
      <atom id="a1" elementType="H" hydrogenCount="0" />
      <atom id="a2" elementType="O" hydrogenCount="2" />
      <atom id="a3" elementType="H" hydrogenCount="0" />
    </atomArray>
    <bondArray>
      <bond atomRefs2="a1 a2" order="1" />
      <bond atomRefs2="a2 a3" order="1" />
    </bondArray>
  </molecule>
</cm1>
```
XML applications

- Mathematical Markup Language (Wikipedia):
  Mathematical Markup Language (MathML) is an application of XML for describing mathematical notations and capturing both its structure and content. It aims at integrating mathematical formulae into World Wide Web pages and other documents. It is a recommendation of the W3C math working group.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE math PUBLIC "-//W3C//DTD MathML 2.0//EN" "http://www.w3.org/Math/DTD/mathml2/mathml2.dtd">
<math mode="display" xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mi>x</mi>
    <mo>=</mo>
    <mfrac>
      <mrow>
        <mo form="prefix">−</mo>
        <mi>b</mi>
        <mo>±</mo>
        <msqrt>
          <msup>
            <mi>b</mi>
            <mn>2</mn>
          </msup>
          <mo>−</mo>
          <mn>4</mn>
          <mo>⁢</mo>
          <mi>a</mi>
          <mo>⁢</mo>
          <mi>c</mi>
        </msqrt>
      </mrow>
      <mrow>
        <mn>2</mn>
        <mo>⁢</mo>
        <mi>a</mi>
      </mrow>
    </mfrac>
  </mrow>
</math>
```
XML applications

\[ x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]

RSS:
- XML application used for content syndication in blogs, newspapers, etc.
- Useful for any site that wants to provide a continuing feed of new information
- Normally, an RSS document is associated to a group of HTML pages
XML applications

```xml
<?xml version="1.0" ?>
<rss version="1.0">
  <channel>
    <title>An example of RSS</title>
    <link>http://www.ua.es/</link>
    <description>This is the description</description>
    <language>en</language>
    <copyright>(c) 2011 Sergio Lujan Mora</copyright>
    <item>
      <title>Title</title>
      <description>Description</description>
      <link>http://www.ua.es/</link>
    </item>
  </channel>
</rss>
```

XML applications

- Open XML Format:
  - Microsoft Office 2007 (.docx, .xlsx, etc.)
XML applications
This is a sample document!!!
XML applications

- Scalable Vector Graphics:
  - Format for describing two-dimensional vectorial graphics
  - Many traditional drawing programs can save SVG files just like their native formats
  - Requires special display software:
    - Microsoft Internet Explorer 9
    - Mozilla Firefox 3+

```xml
<?xml version="1.0"?>
<svg xmlns="http://www.w3.org/2000/svg"
     width="12cm" height="8cm">
  <title>The pink triangle!!!</title>
  <text x="10" y="15">This is SVG!</text>
  <polygon style="fill: pink" points="0,311 180,0 360,311" />
</svg>
```
Exercise
Using SVG make a draw with the following objects: square, rectangle, circle, triangle
Structure of a document

- Logic structure
- Structure of a document
- Definition of a DTD
Logic structure

- XML is based on a containment model:
  - Each XML element (tag) can contain text, other elements, or a mix of both text and other elements

- The first question:
  - What contains what?
  - Which information is a part of which other information?
Structure of a document

- Structure is defined by a DTD (Document Type Definition) or an XML Schema
  - Optional
  - Defines the language: the vocabulary (elements, tags) and the grammar (relationships between elements)

- “Well-formedness”: follows the rules of XML
- “Valid”: follows the rules of a DTD or XML Schema
Structure of a document

- XML declaration → Processing instruction:

```xml
<?xml
    version="1.0"
    encoding="ISO-8859-1"
    standalone="yes" ?>
```

- This declaration must be the first thing in the file
- If this declaration is not included, the default values are:
  - version: 1.0
  - encoding: UTF-8
  - standalone: yes
Structure of a document

- **Comments:**
  - All data inside a comment is ignored by an XML processor
  - Comments cannot come before the XML declaration
  - Comments cannot be placed inside a tag

```xml
<!-- This is a one line comment -->

<!-- This is a multiple lines comment -->
```

Structure of a document

- **Single root element:**
  - An XML document has a root element that completely contains all other elements of the document
  - Root element = document element
Structure of a document

- Tags → Define an element:
  - Delimited by a start-tag (opening tag) and end-tag (closing tag)
  - Every start-tag must have a corresponding end-tag
    `<BOOK></BOOK>`
- Empty tags (start and end tag together):
  `<BOOK/>`
- Elements may nest but may not overlap:
  `<b><i>Some important text</b></i>`

Structure of a document

- Naming conventions of element name:
  - Made up of one or more characters
  - Begin with a letter or an underscore `_`
  - Subsequent characters may include letters, digits, underscores `_`, hyphens `-`, and periods `.`
  - They cannot include white spaces
  - Important: upper- and lowercase are different!
Structure of a document

- Naming conventions:
  - Element names are flexible and can contain any number of letters and digits in either upper- or lowercase
    <countries>
    <Countries>
    <COUNTRIES>
  - It is important to choose one convention and stick to it

- Attributes:
  - Elements can have attributes
  - An attribute is a name-value pair separated by an equal symbol =
  - Attribute names follow the same rules as element names
  - The order inside an element is not important
  - The value of the attribute always between single quotes ' ' or double quotes " "
  - Single quotes and double quotes as values:
    · &quot; and &apos;
Structure of a document

- Predefined attributes:
  - The prefix xml: is reserved for XML specification
  - xml:lang
    - Identifies the language of element’s content
  - xml:space: default | preserve
    - Indicates if the white spaces are significant
  - xml:id
    - Unique identifier in the whole document

Example of xml:lang:

```xml
<p xml:lang="en">The quick brown fox jumps over the lazy dog.</p>
<p xml:lang="en-GB">What colour is it?</p>
<p xml:lang="en-US">What color is it?</p>
<sp who="Faust" desc='leise' xml:lang="de">
  <l>Habe nun, ach! Philosophie,</l>
  <l>Juristerei, und Medizin</l>
  <l>und leider auch Theologie</l>
  <l>durchaus studiert mit heißem Bemüh'n.</l>
</sp>
```
Structure of a document

- White spaces:
  - Tabulator
  - Line feed
  - Carriage return
  - White space

- Normalization of newline characters:
  - Macintosh CR → LF
  - MS-DOS / Windows CR+LF → LF
  - Unix LF

Structure of a document

- Special characters:

<table>
<thead>
<tr>
<th>Carácter</th>
<th>Codificación</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;</code></td>
<td><code>&amp;lt;</code></td>
</tr>
<tr>
<td><code>&gt;</code></td>
<td><code>&amp;gt;</code></td>
</tr>
<tr>
<td><code>&amp;</code></td>
<td><code>&amp;amp;</code></td>
</tr>
<tr>
<td><code>&quot;</code></td>
<td><code>&amp;quot;</code></td>
</tr>
<tr>
<td><code>'</code></td>
<td><code>&amp;apos;</code></td>
</tr>
</tbody>
</table>
Structure of a document

- CDATA sections: XML processor does not try to interpret the content
- Syntax:
  `<! [CDATA [ ...
  ]]>`
- CDATA section cannot be nested or overlapped
- Useful if you want to include large blocks of text that have a lot of &, <, >, ` and "

Exercise

Algorithm example
Structure of a document

```xml
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<algoritmo>
a = 5

si a < 5 entonces
muestra "a < 5"
sino si a > 10 entonces
muestra "a > 10"
</algoritmo>
```

Exercise

Web page
Use W3C markup validator
(http://validator.w3.org/)
Structure of a document

<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>Una prueba del CDATA</title>
<script type="text/javascript">
if(a < 5 || a > 1)
  alert("El valor de a no es correcto");
</script>
</head>
<body>
<p>Una prueba del CDATA</p>
</body>
</html>
Exercise
Write an XML document that stores the information about the books of a library: title, author, editor, number of pages, ISBN, etc.

Other XML technologies
The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential. W3C is a forum for information, commerce, communication, and collective understanding. On this page, you'll find W3C news, links to W3C technology and ways to get involved. New visitors can find help in Finding Your Way at W3C. We encourage you to read the Frequently Asked Questions about W3C.

W3C Requests '905 Patent Re-Examination


W3C Presents W3C Day Japan on 14 November in Tokyo

2003-10-29. W3C Day Japan 2003 (in Japanese) will be held on 14 November 2003 at Keio University Misato Campus.

W3C Holds its First Outreach Event in Mainland China

2003-10-21. The China International Forum on WWWs Development 2003 will be held in Beijing on 12-13 November. Ian Harman, Philipp Hirtle, Richard Ishida, Shizhong Ji, Judy Brewer, and Matthew May present keynotes and tutorials. Attendees will discuss the future of the Web, accessibility, SVG, the mobile Web, the Semantic Web and internationalization. Registration is open. The event is co-organized by the China Computer Federation and the W3C Office in Hong Kong. Read the press release (News archive).

MathML 2.0 Second Edition Is a W3C Recommendation

2003-10-21. The World Wide Web Consortium today released the MathML 2.0 Language (MathML 2.0 Second Edition) as a W3C Recommendation. The specification has been reviewed by the W3C Membership, who favor its adoption by industry. MathML is an XML application that allows mathematical notation and content to be served, received and processed on the Web. The second edition contains clarifications and errata.
Other XML technologies

- XML Namespaces
- XML Schemas
- XPath and XPointer
- XLink, XBase, and XInclude
- XQuery
- XSL/XSLT
- DOM y SAX

Other XML technologies

- XForms
- XMI (*XML Metadata Interchange*)
- XML-QL
- XML Encryption
- XML Signature
- XQL
Software

- Free or pay
- Good place to find programs:
  - [http://www.xmlsoftware.com/](http://www.xmlsoftware.com/)
Software

- **Notepad++**
- Free (GNU General Public Licence)
- General purpose editor, supports many programming languages
- Support of XML is minimum:
  - Syntax highlight
  - Tag coupling
  - Folding (collapsing):
    - This means that certain lines of your text can be hidden based on certain traits
Software

- **Microsoft XML Notepad 2007**
- Free
- Very simple and easy XML editor
- It is not necessary to know how to write an XML document
Software

- ezDTD 1.5
- Free
- DTD editor, both for SGML and XML
- Generates documentation of DTDs in HTML format
Software

- Peter’s XML Editor
  - Free
  - XML and DTD editor
  - Checks well-formedness and validates XML document against DTD
  - Three views of the same document:
    - Source code
    - Tree
    - Internet Explorer
Peter's XML Editor

Versión 2.3

Copyright (C) 1999-2003

This application has been released under the GNU GPL. You should read the terms of the license before using it.

email address: peter@senchusus.org
Website: www.xml.org/poe
Software

- **Cooktop**
- Free
- XML, DTD, XMLSchema, and XSL/XSLT editor
- Checks well-formedness and validates XML document against DTD
- Provides support for XPath and XSL/XSLT
Software

- **EditiX**
- Free and pay version
- Checks well-formedness and validates XML document against DTD
- Special features:
  - DTD ↔ XML Schema conversor
  - SVG viewer
Software

- **XMLSpy**
- Pay
- From company Altova
- Provides support for nearly everything related to XML
- Multiple views of a documents
XML 1.1

- XML 1.0 based on Unicode 2.0:
  - Explicitly listed all the characters than could be used in XML names (element names, attribute names, entity names, and processing instruction targets)
  - Characters not defined in Unicode 2.0 are not allowed in names
    - You can’t write XML 1.0 names in Amharic, Burmese, or Cambodian because those languages (and the corresponding scripts) weren’t added to Unicode until version 3.0
    - But you can use whatever you want in PCDATA (text content)
XML 1.1

- XML 1.1 independent of any particular Unicode version (currently 4.0):
  - Allows to use new characters from new languages (e.g., Burmese, Cambodian, Mongolian, etc.)
  - Allows to use a new newline character (Unicode code 133, NEL)
    - NEL is used as a line terminator on some IBM mainframe systems
  - Forbids the direct inclusion of control characters with Unicode code between 128 and 159 (except NEL)
    - There characters can be included as numeric character references such as &#135; or &#xBC;
  - Allows the inclusion of some additional control characters
  - Includes Unicode normalization

XML 1.1

- Conclusion:
  - W3C: “You are encouraged to create or generate XML 1.0 documents if you do not need the new features in XML 1.1; XML Parsers are expected to understand both XML 1.0 and XML 1.1”
  - You don’t need XML 1.1 unless you need to write elements (markup) in new languages; otherwise, XML 1.1 makes documents incompatible with the large amount of XML 1.0 software
Schemas are documents that define the valid contents of particular classes of XML documents:

- Document Type Definition (DTD)
  - Simple and easy to learn
- W3C XML Schema
  - Large and complex specification
Introduction DTD

- Other popular schema languages:
  - RELAX NG
  - Schematron
- Dead schema languages:
  - Document Content Description (DCD)
  - Commerce One’s Schema for Object-Oriented XML (SOX)
  - Microsoft XML-Data Reduced (XDR)
  - ...

Definition of a DTD
Definition of a DTD

- Document Type Definition (DTD) defines the language (vocabulary + grammar) of an XML document:
  - It lists the elements, attributes, entities, and notations that can be used in a document
  - It defines the possible relationships between the elements

- XML is based on a containment model:
  - Each XML element (tag) can contain text, other elements, or a mix of both text and other elements
  - The first step to creating a DTD is to understand the structure of the information:
    - What contains what?
    - Which information is a part of which other information?
Definition of a DTD

- Each rule is a declaration: 
  `<! ... >`
- Four types of declarations:
  - Element (ELEMENT)
  - Attribute (ATTLIST)
  - Entity (ENTITY)
  - Notation (NOTATION)
Definition of a DTD

- Comments:
  - Similar to an XML document
  - All data inside a comment is ignored by an XML processor
  - Comments cannot be placed inside a declaration
  
    <!-- This is a one line comment -->

    <!-- This is a multiple lines comment -->

Element

- Defines an element (tag):
  
    <!--ELEMENT name contentSpecification-->

- Name:
  - First character: letter, `_`, `\`
  - Following characters: letters, `_`, `\`, digits, `.` o `\`

Máster Web
## Element

- **Content specification**: determines what an element may and may not contain
  - Nothing: **EMPTY**
  - No restrictions on the content: **ANY**
  - Children: **content model**
  - Text: **#PCDATA** (*parsed character data*)
  - Mix

## Element

- **Content model**: 
  - A parenthesized list of child elements
  - Sequence: `,`
    - A list of child elements separated by commas
    - Elements appear or do not appear in a specific order
  - Choice: `|`
    - A list of child elements separated by vertical bars
    - Exactly one child element of a group of child elements to appear
Element

- Content model:
  - Repetition:
    - + One or more children
    - ? Zero or one child
    - * Zero or more children
  - Parentheses: ( )
    - Each set of parentheses combines several elements so that the combination is treated as a single unit when validating
    - Sequences and choices can be nested to produce more complex content models

Exercise
Propose valid and no valid XML fragments for the following declarations
Exercise

<!ELEMENT x (a, b?, c+, d*)>
<!ELEMENT x ((a | b), c?)>
<!ELEMENT x ((a | b)+, c?)>
<!ELEMENT x ((a, b)+, (c | d)+)>

Exercise

Propose two valid declarations for the following XML fragment
Exercise

<x>
<a></a>
<b></b>
<c></c>
</x>

Element

- Mixed content:
  - Contains both child elements and parsed character data
  - Structure very restrictive:
    - `#PCDATA` always the first element
    - Choice: (1)
    - Zero or more (*)
  - Example:
    ```
    <!ELEMENT parent (#PCDATA | child1 | child2 child3) *>  
    ```
Attribute

- Defines attributes of an element (tag):
  - Attribute: name-value pair separated by an equals sign (=)

```xml
<!ATTLIST element
  attributeName1 type1 defaultValue1
  attributeName2 type2 defaultValue2>
```

### Attribute Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDATA</td>
<td>Character data, text that is not markup</td>
</tr>
<tr>
<td>NMTOKEN</td>
<td>An XML name token</td>
</tr>
<tr>
<td>NMTOKENS</td>
<td>Multiple XML name tokens separated by white spaces</td>
</tr>
<tr>
<td>Enumerated</td>
<td>A list of possible values from which exactly one will be chosen</td>
</tr>
<tr>
<td>ID</td>
<td>A unique name not shared by any other ID type attribute in the document</td>
</tr>
<tr>
<td>IDREF</td>
<td>The value of an ID type attribute of an element in the document</td>
</tr>
<tr>
<td>IDREFS</td>
<td>Multiple IDs of elements separated by white spaces</td>
</tr>
<tr>
<td>ENTITY</td>
<td>The name of an unparsed entity declared in the DTD</td>
</tr>
<tr>
<td>ENTITIES</td>
<td>Multiple names of unparsed entities declared in the DTD, separated by white spaces</td>
</tr>
<tr>
<td>NOTATION</td>
<td>One or more names of notations declared in the DTD</td>
</tr>
</tbody>
</table>
Attribute

Default value alternatives:
- #REQUIRED → The attribute is mandatory
- #IMPLIED → The attribute is optional
- #FIXED "value" → The attribute has a fixed value and cannot be changed
- "valor" → A particular default value

<!ELEMENT AUTHOR EMPTY>
<!ATTLIST AUTHOR NAME CDATA #REQUIRED>
<!ATTLIST AUTHOR TELEPHONE CDATA #IMPLIED>
<!ATTLIST AUTHOR EMAIL CDATA #REQUIRED>
<!ATTLIST AUTHOR COUNTRY CDATA "Spain"uellement
Attribute

```xml
<!ELEMENT AUTHOR EMPTY>
<!ATTLIST AUTHOR
  NAME CDATA #REQUIRED
  TELEPHONE CDATA #IMPLIED
  EMAIL CDATA #REQUIRED
  COUNTRY CDATA "Spain"
>
```

- **NMTOKEN** *(name token)* similar to CDATA, restricts the value of the attribute to a legal XML name token:
  - A name token must contain letters, digits, underscores, hyphens, and periods
  - No white spaces
- **NMTOKENS**: multiple NMTOKEN separated from each other by white spaces
Attribute

- Enumerated:
  - A list of possible values separated by vertical bars
  - Each value must be a valid XML name token

```xml
<!ATTLIST p visible (true | false) "true">
```

```
<p visible="false">This paragraph is not visible!</p>
```

Attribute

- ID:
  - Uniquely identifies an element in the document:
    - Using the same ID value twice in one document causes an error
  - The value must be a valid XML name:
    - Must begin with a letter and is composed of alphanumeric characters and the underscore without white spaces
Attribute

- **IDREF:**
  - References the ID value of another element in the document
  - IDREF and ID attributes are used to establish connections between elements that aren’t reflected in the tree structure of the document

- **IDREFS:**
  - Contains a list of ID values separated by white spaces

Attribute

- **ENTITY:**
  - Contains the name of an unparsed entity declared in the DTD
  - Normally used to link external binary data

- **ENTITIES:**
  - Contains a white-space separated list of unparsed entity declared in the DTD
Attribute

- NOTATION:
  - Contains the name of a notation declared in the DTD
  - A notation identifies the format of data

```xml
<!NOTATION ISODATE SYSTEM "http://www.iso.ch/cate/dl5903.html">
<!NOTATION USDATE SYSTEM "http://tf.nist.gov/timefreq/general/enc-d.htm#date">

<!ATTLIST DATE FORMAT NOTATION (ISODATE | USDATE) #IMPLIED>
```

Example

```xml
<!ATTLIST img
  src CDATA #IMPLIED
  align (left | center | right) "left"
  size NMTOKEN #REQUIRED>

<!ATTLIST student
  name CDATA #REQUIRED
  number ID #REQUIRED
  course CDATA #FIXED "First">
```
Example

```xml
<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE AGENDA [ 
<!NOTATION ISODATE SYSTEM "http://www.iso.ch/cate/d15903.html"> 
<!NOTATION USDATE SYSTEM "http://tf.nist.gov/timefreq/general/enc-d.htm#date"> 
<!ELEMENT AGENDA (CITA*)> 
<!ELEMENT APPOINTMENT (MESSAGE, DATE, TIME?)> 
<!ELEMENT MESSAGE (#PCDATA)> 
<!ELEMENT DATE (#PCDATA)> 
<!ELEMENT TIME (#PCDATA)> 
<!ATTLIST DATE FORMAT NOTATION (ISODATE | USDATE) #IMPLIED> ]>

Example

<AGENDA>
  <APPOINTMENT>
    <MESSAGE>Attend the XML course</MESSAGE>
    <DATE FORMAT="USDATE">11-10-2009</DATE>
  </APPOINTMENT>
  <APPOINTMENT>
    <MESSAGE>Finish exercise of the XML course</MESSAGE>
    <DATE FORMAT="ISODATE">20091117</DATE>
    <TIME>23:59</TIME>
  </APPOINTMENT>
</AGENDA>
```
Entity

- An entity holds content: well-formed XML, other forms of text, or binary data
- Advantages:
  - Helps to write documents
  - Reduces errors
  - Clarifies documents
  - Helps to update documents
- Different types:
  - Internal or external
  - Parsed or unparsed

Internal:

- `<!ENTITY entityName "replacementValue">`
- `&entityName;`

External:

- `<!ENTITY entityName SYSTEM "file.xml">`
- `&entityName;`
Entity

- Predefined entities:
  ```xml
  <!ENTITY lt "&#38;#60;">  
  <!ENTITY gt "&#62;">  
  <!ENTITY amp "&#38;&#38;">  
  <!ENTITY apos "&#39;">  
  <!ENTITY quot "&#34;">
  ```

Entity

- Parameter entity:
  - Associates a name with a DTD fragment
  - Parameter entity references can only appear in the DTD, not in the document
  ```xml
  <!ENTITY % name "replacementValue">
  ```

- How to use:
  ```xml
  %name;
  ```
Example

<!ELEMENT img EMPTY>  
<!ATTLIST img
%attrs;
src   %URI;       #REQUIRED
alt   %Text;      #REQUIRED
longdesc %URI;    #IMPLIED
height %Length;   #IMPLIED
width  %Length;   #IMPLIED
usemap %URI;      #IMPLIED
ismap (ismap)     #IMPLIED
>

<!ENTITY % URI "CDATA">
<!ENTITY % Text "CDATA">
<!ENTITY % Length "CDATA">

Example

<!ENTITY % TextAlign "align (left|center|right|justify)
#IMPLIED">

<!ATTLIST div
%attrs;
%TextAlign;
>

<!ATTLIST p
%attrs;
%TextAlign;
>

<!ATTLIST h1
%attrs;
%TextAlign;
>
**Notation**

- A notation describes one possible format for non-XML data.
- Notations are declared in the DTD and then used as the values of `NOTATION` attributes.
- It is not well supported, therefore, it is not used.

**Example:**
- Defines the format of GIF files

```xml
<!NOTATION GIF SYSTEM "GIF">
<!NOTATION GIF PUBLIC "image/gif">
<!NOTATION GIF PUBLIC "-//IETF//NONSGML Media Type image/gif//EN">
```
Document Type Declaration

- A document type declaration is placed at the beginning of an XML document
  - Document Type Declaration is not the same thing as Document Type Definition (DTD)
- It specifies which element is the root element of the document

Document Type Declaration

- The DTD can be included in the XML document (internal DTD) or the XML document can link to it (external DTD)
- An external DTD can be shared by different documents
Document Type Declaration

- At the beginning:
  - Internal DTD
    ```xml
    <!DOCTYPE library [...]>
    ```
  - External DTD
    - By convention, extension .dtd
    - PUBLIC: global, standard
      ```xml
      <!DOCTYPE rootElement PUBLIC "DTDname">
      ```
    - SYSTEM: local, defined by the user (private)
      ```xml
      <!DOCTYPE rootElement SYSTEM "DTDurl">
      ```
  - A combination (internal + external)

Example:

```xml
<!DOCTYPE GREETING [  
<!ELEMENT GREETING (#PCDATA)>  ]>
</!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!DOCTYPE library SYSTEM "library.dtd">
<!DOCTYPE DOCUMENT SYSTEM "greeting.dtd" [  
<!ELEMENT DOCUMENT (GREETING, DATE)>  
<!ELEMENT DATE (#PCDATA)>  ]>
```
Processing order

```xml
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
```

Validation of a DTD

- **Validome:**
  - [http://www.validome.org/grammar/](http://www.validome.org/grammar/)
  - Validates a DTD or XML Schema
Validation of a DTD

- **EditiX:**
  - Validates DTD syntax
Popular DTDs
Popular DTDs

- The real power of XML comes from common DTDs that are shared among many users
- DTDs can ensure that different people and programs can read each other’s documents

Popular DTDs

- Book XML DTD
- Channel Definition Format (CDF)
- Chemical Markup Language (CML)
- Genealogical Data in XML (GedML)
- Mathematical Markup Language (MathML)
- Open Software Description (OSD)
- Resource Description Framework (RDF)
- Web Distributed Data Exchange (WDDX)
This document type definition (DTD) models the Fleet Technical Support Center Planned Maintenance System (PMS) document classes. It is invoked by the document type declaration:

```xml
<!DOCTYPE docpms PUBLIC "-//USA-DO/DTD FOR FTSC PMS DOCUMENTS 990729/EN">```

This DTD may be revised as the FTSC PMS automation effort continues. For information regarding the PMS Program and the DTD’s availability, contact:

Mr. Jim Helms
FTSCLANT Norfolk VA, Code 4103
jim_helms@ftsclant.navy.mil
(757) 485-6112

For information regarding the DTD’s application and use, contact:

Mr. Jim Pope
FTSCPAC San Diego CA, Code 401
popej@mailhost.ftscpac.navy.mil
(619) 524-2536

This DTD was written for FTSC by the Technology Implementation Support Team of the Naval Surface Warfare Center, Carderock Division, David Taylor Model Basin. For information regarding the development of this DTD, contact either:

Ms. Lori Westbrook
westbrookl@nsnwcdd.navy.mil
VOICE: 301-227-3348
FAX: 301-227-3348

Mr. Donald Gignac
gignaca@nsnwcdd.navy.mil
VOICE: 301-227-3348
FAX: 301-227-3348

at the
Technology Implementation Support Team (Code 2053)
Naval Surface Warfare Center
Carderock Division
9500 MacArthur Boulevard
West Bethesda, MD 20817-5700.
Validation of a document

- **Well-formedness** → XML rules
  - The XML declaration
  - Single root element
  - Every start-tag must have a corresponding end-tag
  - Elements may nest but may not overlap
  - ...  
- **Valid** → DTD rules
  - A document can be compared against a DTD in a process known as validation
No se puede mostrar la página XRI.

No se puede ver la entrada XRI con la hoja de estilo XSL. Corrige el error y vuelve clic en el botón Actualizar, o intentalo de nuevo más tarde.

Se esperaba una cadena literal, pero no se encontraron las cadenas de apertura. Error en procesar el recurso:

<?XRI?><!DOCTYPE CSS[<!ELEMENT XRI (xri) ]>

No se puede mostrar la página XRI.

No se puede ver la entrada XRI con la hoja de estilo XSL. Corrige el error y vuelve clic en el botón Actualizar, o intentalo de nuevo más tarde.

La etiqueta de fin 'XRI' se coincide con la etiqueta de inicio 'XRI'. Error en procesar el recurso:

'XRI'>
Validation of a document

To validate a document against its DTD in Microsoft Internet Explorer we need to install **Internet Explorer Tools for Validating XML and Viewing XSLT Output**
Validation of a document

- **EditiX:**
  - Validates XML documents against DTDs