Practical Use of XML

Rostislav Titov

IT-AIS-EB (e-Business) Section
CERN – Geneva, Switzerland
XML

eXtensible Markup Language

- SGML (ISO standard, 1986)
  Mainly for technical documentation
- XML (W3C recommendation, 1998)
  Simplification and enhancement of SGML, wide area of use
Why Markup?

Markup allows to add information about data structure.
**XML: Rules**

- **Header**
- **One root element**
- **Tag hierarchy**
- **Attributes**
- **Text elements**
- **Empty elements**

**Some rules**

- Element names are case-sensitive
- Every opening tag should have a closing tag
- Tags cannot intersect (<a><b></a></b>)
- Attributes values – in quotes or apostrophes

```xml
<?xml version="1.0" encoding="UTF-8"?>
<presentation>
  <author>
    <firstname>Rostislav</firstname>
    <lastname>Titov</lastname>
  </author>
  <chapter number="1" title="What is XML">
    XML (Extensible Markup Language) is …
  </chapter>
  <conclusion/>
</presentation>
```
XML: Data Transfer

- Platform and language independent
- Easy to write, easy to process
- Understandable for humans and computers
- Open standard
  - Many libraries exist
  - Lots of literature available
  - Specialized XML-editors
- Possibility to check the document structure
XML: Data Transfer (2)

Example: EDH Transport Request

- Automatic form generation from external programs
- XML as data transfer format
- Schema checkup as a warranty of data consistency
Web Services

- Data transfer between programs on Internet
- Open Standard
- Platform and language independent (Java, .Net, …)

Web service

XML

WSDL

SOAP

WSDL – Web Service Definition Language

SOAP – Simple Object Access Protocol
XML: Data Storage

- Data structure is kept together with the data
- Object “addendum” to relational RDBMS
- Structure checkup
- Supported by many modern RDBMS
  - Microsoft SQL Server 2005, Oracle 9i +,
  - XML Data Type
  - XML indexes
  - XML Queries (XQuery etc.)
  - Data output in XML format
Example: EDH Search System

Problem: Effective search using arbitrary number of criteria is problematic

Our solution:

- All documents are stored in XML
- Context-specific XML search (Oracle InterMedia)

Example: «Find documents created by Slava»:

Select DOC_ID from DOC_XML where Contains(XML, “Slava within creator”) > 0;
XML: Data Transformations

- XML can be transformed into HTML, text, PDF, ...  
  - No need for special program solutions
  - Commercial visual editors exist
  - Platform independent
XML-based Standards

- Possibility to formally define the structure
- Platform and language independent
- Understandable for humans and computers
- Possibility to use XML technologies (XSLT transformations, XQuery queries)...
  - WSDL (Web Services Definition Language)
  - SOAP (Simple Object Access Protocol)
  - XHTML (HTML that complies to XML rules)
  - SVG (Scalable Vector Graphics)
  - ebXML (XML for e-Business)
  - ...
Formal Structure Definition

• There are ways to define XML structure formally

  • DTD (Document Type Definition) **Obsolete! Not for new development**
  • XML Schema
XML Schema: Possibilities

- Check element presence and their order
- Sequences and choices
- Number of repetitions for elements and groups
- Attributes and their presence
- Type of elements and attributes
- Restrictions for elements and attributes
- Default values
- Unique constraints
- ...
XML-schema: when it is needed?

- Formal structure definition for future reference
- Programmers may rely on data consistence
- Authors may check XML validness in advance
When we know in advance that XML is valid

When we do not care about document validness

When maximum processing speed is required

Small “throw away” projects
XPath: XML Navigation

- Access to XML elements

- Result of an XPATH-expression can be:
  - XML Node
  - Node Set
  - Boolean
  - String
  - Number
  - Empty Set

C:\presentation\author\firstname  /presentation/author/firstname

XML Node

Node Set

Boolean

String

Number

Empty Set
XPath: Examples

- Find the DG’s name
  `/cern/dg/person/text()`

- Find all departments
  `/cern/department/@name`

- Find all people
  `//person`

- Find the name of DH of IT
  `/cern/department[@name="IT"]/dh/person/text()`

- Find how many groups has a department where R. Martens works
  `count(/gl/person[starts-with(., 'R. Martens')]/../../group)`
Example: Event Handling System

«I want to see all documents for more than 600 CHF»

/ document [amount > 600]
XPath: Program Use

**XPath**

```java
Element root = xml.getDocumentElement();
Node child;
for (child = root.getFirstChild(); child != null; child = child.getNextSibling())
    if (child.getNodeName().equals("report") && ((Element)child).getAttribute("name").equals("Slava"))
        break;
for (child = ((Element)child).getFirstChild(); child != null; child = child.getNextSibling())
{
    if (child.getNodeName().equals("title"))
    {
        for (Node child2 = child.getFirstChild(); child2 != null; child2 = child2.getNextSibling())
            if (child2 instanceof Text)
                System.out.println(((Text)child2).getData().trim());
    }
}
```

**DOM Model**

```xml
<config>
  <report name="Vasya">
    <author>X</author>
    <title>Vasya's report</title>
  </report>
  <report name="Slava">
    <author>Y</author>
    <title>Slava's report</title>
  </report>
</config>
```
XQuery – XML Query Language

- XQuery is SQL for XML
  - Database independent
  - Easy to use

- Supported by popular RDBMS
  (Microsoft SQL Server 2005, Oracle 9i and 10g)

- Based on XPath, supports document sets
XSLT: XML Transformations

- Transforms XML to HTML, text or other XML
- **XSLT 1.0 (Current), XSLT 2.0 (Draft)**
- XSLT is a “Human Interface” to XML
- Supported by Web Browsers

Practical Use of XML

**Author:** Rostislav Titov

**Table of Contents**

1. What is XML
2. XML Structure

**Chapter 1. What is XML**

XML (Extensible Markup Language) is a standard, proposed by the W3C consortium in 1996.

**Chapter 2. XML Structure**

XML is a normal text file that could be edited in any text editor, such as NotePad.
XSLT: Simplified Structure

- XSLT is an XML file
- Active usage of XPath expressions

```
<html>
  <body>
    ...
  </body>
</html>
```

- `xsl:stylesheet`<br>
- `xsl:template`<br>
- `xsl:value-of`<br>
- `xsl:apply-templates`<br>

Apply a template to the given element<br>
Evaluate XPath and print value<br>
Apply templates to other elements
XSLT: Possibilities

- Conditions (<xsl:if>)
- Loops (<xsl:for-each>)
- Variables (<xsl:variable>)
- Sorting (<xsl:sort>)
- Numbering [1., 1.1., 1.1.?, 2.,] (<xsl:number>)
- Number formatting (format-number())
- Multiple step processing (mode)
- String manipulations (via XPath)

XSLT 2.0 (Draft)
- XPath 2.0
- Custom functions
- Regular expressions
- Date and time formatting
- Groupings
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" version="1.0" encoding="UTF-8" indent="yes"/>

  <xsl:template match="presentation">
    <html>
      <body bgcolor="#FFCCFF">
        <h1><font color="darkblue"><xsl:value-of select="title"/></font></h1>
        <h4><font color="green"><i>Author: <xsl:value-of select="author"/></i></font></h4>
        <b>Table of Contents</b><br/><br/>
        <xsl:apply-templates select="chapter" mode="contents"/>
        <br/><br/>
        <xsl:apply-templates select="chapter" mode="normal"/>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="chapter" mode="normal">
    <b>Chapter <xsl:value-of select="@number"/>. <xsl:value-of select="@title"/></b><br/><br/>
    <i><xsl:value-of select="text()"/></i><br/><br/>
  </xsl:template>

  <xsl:template match="chapter" mode="contents">
    <xsl:value-of select="@number"/>. <xsl:value-of select="@title"/>
  </xsl:template>
</xsl:stylesheet>
XSLT: Web “Skins”

```xml
<aissearchscreen>
  <head><title>Person Search</title></head>
  <body>
    <input type="hidden" name="isAdvanced" value="false"/>
    <input show="always" type="text" label="Keyword" value="titov"/>
    <input type="checkbox" label="Fuzzy search" value="No"/>
    <result>
      <header>
        <tablecell>Full Name</tablecell>
        ...
      </header>
      <row>
        <tablecell>Maksym TITOV</tablecell>
        <tablecell>71169</tablecell>
        <tablecell>403-C08</tablecell>
        ...
      </row>
      <row>
        <tablecell>Oleg TITOV</tablecell>
        <tablecell>EXT</tablecell>
        ...
      </row>
      <rowcount>4</rowcount>
    </result>
  </body>
</aissearchscreen>
```
CERN Stores Catalog

- Data loaded through XML
- Data stored in XML
- XSLT for data output
- 150000 items
- +10000 users
- ~15-20K XML for each page
- Custom formatting (through XSLT redefinition)
**Example:**

- **Automatic code generation**

```xml
<document>
  <input type="person" name="A"/>
  <input type="number" name="B"/>
  ...
</document>
```

**Did you know...**

that 1 EDH document is:

- At least 20 source files (code, HTML templates, resources, SQL, ...)
- About 250K of source code
XSLT: XML to XML

- Generate XML from another XML source
- “Configuration files update”
- XSL:FO
XSL-FO: Formatting Objects

- FO: XML-description of document layout
- XSL-FO: XSLT transformation of XML document to FO document
- FO Processor: program that converts the FO definition into a printable format (PDF, PS, ...)

XML Document

```
<?xml version="1.0"?>
<presentation>
  <title>XXX</title>
</presentation>
```

FO Document

```
<fo:root>
  <fo:page-sequence>
    <fo:flow>
      ...
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

PDF Document

```
<?xml version="1.0"?>
<presentation/>
```
XSL-FO: Formatting Objects

*FO has all capabilities of modern text editors:*

- Fonts
- Pagination
- Headers and footers
- Page numbering
- Odd/even page distinction
- Margins and intervals
- Keep paragraphs together
- Hangout lines
- Tables
- Graphics
- ...

**FO Processor:**
Apache FOP
XSL-FO: Example

- No extra code required
- RTF to XSL:FO converters are good
- Can be written by a student
- Output format independent
XML Editors

- Specially designed for XML editing
- XML well-formedness and validity check
- DTD and Schema visual editing
- XML generation accordingly to DTD/Schema
- Creation and debugging of XSLT and XSL:FO
- Visual XSLT editing

Example: Altova XML Spy (www.xmlspy.com)

- Available from NICE
- License can be obtained from the SDT service
XML: Program Handling

- **DOM (Document Object Model)**
  - Tree building

- **SAX**
  - Event handling
    - `startElement()`
    - `endElement()`

  **SAX** - much faster,
  **DOM** – more versatile

**Java, C++:**
- Apache Xalan
- Oracle XML Parser

**PERL, .Net:**
- Built-in support
New Technologies

- **InfoPath 2003**
  - Corporate system for electronic form handling
  - XML-based
  - Business rules defined by XML schema
  - Data validation using XML schemas

- **Adobe Intelligent Document Platform**
  - Similar ideas
Conclusion

«XML is one of the biggest inventions in IT area in the last few years. There is a lot of XML applications around the world today, and this amount will grow every year»

W3C Consortium Web Site:
http://www.w3c.org

Questions:
Rostislav.Titov@cern.ch