

Practical XML/XSL Solutions

Practical XML/XSL Solutions for an Electronic Business



Overview

- Generic business model (15 minutes)
- XML/XSL In depth (20 minutes)

World Wide Widgets

- Our Goals:
 - Make the best widgets in the world
 - Ship worldwide



Use XML on the Backend

- The most basic description of our business:
an order:



Basic XML Order

```
<?xml version="1.0" ?>
- <orders vendorid="1" version="1">
- <order>
    <orderid>1234</orderid>
+ <vendor logoimageid="3">
+ <customer>
+ <shippingaddress>
    <shippingmethod>FC</shippingmethod>
+ <orderline productid="10080">
    <producttotal>13.00</producttotal>
    <shippingprice>3.95</shippingprice>
    <discount>0.95</discount>
    <subtotal>16.00</subtotal>
    <tax>0.00</tax>
    <ordertotal>16.00</ordertotal>
    <coupon imageid="" />
    <comment>Good Luck</comment>
  </order>
</orders>
```

Supporting Documents

- DTD/Schema
 - Multiple Formats out there
 - W3C Schema in Candidate Recommendation Status
 - <http://www.w3.org/2000/10/XMLSchema>
 - Use this to define your namespace

Interfaces

- Our XML documents becomes our main electronic interface with partners
 - Receive orders from partners
 - Add status information/Comments and send back to partners
- Use XSL to create various visual interfaces
 - HTML, PDF, etc.

XSL: XSLT & XSL FO

- Use XSLT to create Web Interfaces (HTML, WML, etc)
 - Internal Reports:
 - Order Summary & customer service tools
 - Production & support tools
 - Shipping & receiving tools

- External reports
 - Customers & partners
- Use XSL FO to create printable reports
 - Invoices/Packing slips
 - Charts & Graphs (SVG)
 - Barcodes

Use XML on the Front-end

- Used in our shopping cart to enhance Partner opportunities
 - Each page has a schema which includes part of the Final order
 - Customer, Shipping data, line items
- With XSLT partners can create custom versions of our cart that we host

The good news and the bad news

- XML is the perfect standard interface for conducting e-business.
- Tools are scarce, especially for namespaces that are not finalized
 - XSLT: Extensible Style Sheet Transformations
 - XSL FO: Formatting Objects
 - SVG: Scalable Vector Graphics

In Depth: Algorithms with XSL

- Problem:
 - We have an extensive list of products and a complex shipping price algorithm which we use to bill partners.
 - Partners want to know at the time an order is placed what the shipping cost will be
 - Partners want autonomy from our systems

- Solution:
 - Using XSL, we can implement a transformation that modifies our XML order document to include shipping costs.
 - XSLT solves both problems, if we can implement it!

- Other possible solutions: SOAP, Java
 - Advantages of XSLT
 - XSLT is easier to develop for multi platform use
 - Can be developed and modified easily
 - Disadvantages:
 - XSLT is not an algorithm friendly language
 - Little support for data types. It is designed more to extract and transform data.

Getting Started

- Step 1: Gather the data
 - XML order document
 - Product information
 - Shipping information
 - `<xsl:variable name="shippingData" select="document('shippingdata.xml')"/>`

What is my context?

- Context changes as you drill deeper into a document.
- If you switch documents, context changes to that document
- Use variables to keep your documents straight

Variables & Data Types

- Variables:
 - XSL variables are actually more like place holders
 - You can't change the value of an `<xsl:variable/>`
 - Useful in xpath and for storing a node

Statements & Built in functions

- Looping:
 - <xsl:foreach/>
 - <xsl:apply-templates/>
 - <xsl:sort/>
- Decisions
 - <xsl:if/>
 - <xsl:choose/>

XPath

- Sample String Function
 - String-length ()
 - Contains ()
- Sample Numeric function
 - Count ()
- Sample Axes
 - Following-sibling

Reference

- www.w3.org
- FAQ assembled by Dave Pawson:
<http://www.dpawson.co.uk/xsl/xslfaq.html>
- Microsoft's MSXML 3.0 SDK:
<http://msdn.microsoft.com/xml>

