

# eXtended Markup Language (XML) Overview

Kelvin R. Lawrence

IBM Distinguished Engineer &  
CTO , Emerging Internet Software Standards

IBM Software Group

klawrenc@us.ibm.com

[http://www.ibm.com/developerworks/blogs/dw\\_blog.jspa?blog=730](http://www.ibm.com/developerworks/blogs/dw_blog.jspa?blog=730)

# Agenda

## Core XML Technologies

- ▶ XML language structure
- ▶ DTDs and Schemas
- ▶ XSL (XML stylesheets)
- ▶ XML standards summary
- ▶ A brief word about Web Services

# What is XML?

# What is XML?

- Tags in text to mark-up meaning
- A text-based tag language, similar in style to HTML, but lets you define your own tags
- A standard way of sharing structured data
  - ▶ A key technology to enable e-business
- A simplified subset of SGML
- A language for defining other markup languages, interchange formats and message sets

# How is XML used?

## ■ Documents

- ▶ purchase order, employee record, electronic Trading Partner Agreements, structured text documents, ...
- ▶ common import/export format
- ▶ great for integrating heterogenous applications

## ■ Messages

- ▶ service request  
(example: "please verify this credit card #")

# Why XML is so Important

- **I**nformation
  - ▶ messages and documents
- **I**nteroperability
  - ▶ sharing data across applications and platforms
- **I**ntegration
  - ▶ bringing together data from multiple sources
- **I**ndependence
  - ▶ application, programming language, operating system, delivery device, hardware platform
- **I**nternationality
  - ▶ designed to use UNI CODE

# Sample XML code

```
<address>
  <name>
    <title>Mrs.</title>
    <first-name>Mary</first-name>
    <last-name>McGoon</last-name>
  </name>
  <street>1401 Main Street</street>
  <city>Sheboygan</city>
  <state>WI</state>
  <zip>38472</zip>
  <country>USA</country>
</address>
```

## Compared to HTML, XML

- ▶ labels the data - says what it is
- ▶ does NOT say how it should be presented

# Well-formed XML

A small set of rules define the basic syntax that any XML document must follow

- ▶ required first line `<?xml version="1.0"?>`
- ▶ syntax of tags `<tag>data</tag>`
- ▶ nesting of tags 

```
<employee>
  <name>Mark Colan</name>
  <id>X04913</id>
</employee>
```
- ▶ tag attributes `<tag attribute="x" />`

A "valid" XML document is well-formed AND complies to specific DTD or XML Schema

# XML rules: closing tags required

- Legal HTML, not legal in XML:

```
<p>Explain this!  
<br>
```

- XHTML: legal for both XML and HTML:

```
<p>Explain this!</p>
```

- Legal XML shortcut for a tag with no text data:

```
<br />
```



Note the slash before closing bracket.

# XML rules: correct nesting

- HTML can be written to be conformant XML.
- This is legal in HTML, but illegal in XML:


```
<p>Explain <b><i>this</b></i>!
```

- XHTML: legal for HTML and XML

```
<p>Explain <b><i>this</i></b>!</p>
```


This is legal:

```
<i>
  <b>
  </b>
</i>
```



This isn't:

```
<i>
  <b>
</i>
  </b>
```



# XML rules: outer tag set

The XML document must be enclosed in one set of tags.

This is legal:

```
<colors>  
  <color>red</color>  
  <color>green</color>  
</colors>
```

This isn't:

```
<color>red</color>  
<color>green</color>
```

# XML attributes

You can use attributes within a tag:

```
<paper color="red">  
  Print this on red paper.  
</paper>
```

Maybe there's no data. Use either form:

```
<paper color="red"></paper>  
<paper color="red"/>
```

The second form is a short-cut for the first; the two are equivalent.


# XML for specific data sets

- What we have seen so far is the rules that any XML document must follow.
- How do we specify:
  - ➔ the names of tags we allow?
  - ➔ which tags can nest other tags?
  - ➔ required vs optional for each tag?
  - ➔ one occurrence, or any number?
  - ➔ default value of attributes?

# XML Vocabularies

- A particular XML markup language is called a "vocabulary", expressed in either or both:
- DTD (Document Type Definition)
  - ➔ part of the XML 1.0 specification
  - ➔ comes from SGML definition
- XML Schema
  - ➔ an improved XML definition language
  - ➔ recommendation from W3C
  - ➔ may 2001
- We'll look at DTDs and Schemas later.

# XML Summary

- XML is data annotated by meaningful tags
  - presentation is separate; specified by stylesheet
  - data is easily accessed by a program
  - is perfect for structured data interchange
  - great for application integration
  - independent of programming language, platform, delivery device, operating system, ...
  - international
-  **XML is a key component for e-business.**

# DTDs: Document Type Definitions

# Document Type Definitions

- The structure of an XML document is defined by its DTD. DTDs define:
  - ▶ the tags that can or must appear
  - ▶ how often the tags can appear
  - ▶ how the tags can be nested
  - ▶ allowable, required and default attributes
- But note: the use of DTDs is optional!
  - ▶ DTD allows a validating parser to detect deviations from a vocabulary
  - ▶ Can parse well-formed XML without a DTD

# Sample XML code

```
<address>
  <name>
    <title>Mrs.</title>
    <first-name>Mary</first-name>
    <last-name>McGoon</last-name>
  </name>
  <street>1401 Main Street</street>
  <city>Sheboygan</city>
  <state>WI</state>
  <zip>38472</zip>
  <country>USA</country>
</address>
```

# Sample DTD

Outer group <address>

```

<!ELEMENT address
  (name, street*, city,
  state, zip?, country)>
<!ELEMENT name
  (title?, first-name,
  last-name)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT first-name (#PCDATA)>
<!ELEMENT last-name (#PCDATA)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT zip (#PCDATA)>
  
```

May have one or more <street> tags

Must have one each: <name>, <city>, <state>, <country>

<title> and <zip> are optional

may appear at most once

any text: alpha, numeric, punctuation, whitespace

# What's wrong with DTDs?

- No type support - **#PCDATA** can be any string of characters (except tags)
- DTD syntax is different from XML syntax
  - <!ELEMENT zip (#PCDATA)>**
- DTDs cannot express specific constraints:
  - ▶ element x can occur from 4 to 17 times
  - ▶ if the type of element y is "decimal", the y element must contain an x element
- But, as Yoda said, "There is... another."

# XML Schema

# XML Schema

## An improved XML vocabulary definition language:

- ▶ written in XML syntax (DTDs are not)
- ▶ superset of DTD functionality
- ▶ Schema composition
- ▶ namespace support

## Biggest improvement is specification of types

- ▶ Built-in Simple Types
- ▶ Derived types ("subclass" of built-in Simple Types)
- ▶ Complex types: definition of an item with sub-items

# Built-in Simple Types

- string, boolean
- float, double,
- decimal
- timeInstant
- timePeriod
- timeDuration
- month, year,
- century
- recurringDate, recurringDay,
- recurringDuration
- integer, nonPositiveInteger, positiveInteger, nonNegativeInteger, negativeInteger, long, int, short, byte, unsignedLong, unsignedInt, unsignedShort, unsignedByte
- uriReference
- date
- time
- language
- Name
- QName
- NCName
- ID
- IDREF
- IDREFS
- ENTITY
- ENTITIES
- NOTATION
- NMTOKEN
- NMTOKENS

# Simple and Complex types

```
<shoeOrder ship='2000-05-16'>  
  <size>9</size>  
</shoeOrder>
```

- **<size> is a Simple Type**
  - ▶ integer type is "built-in" to schema
  - ▶ no sub-elements or attributes
- **<shoeOrder> is a Complex Type**
  - ▶ has an element
  - ▶ has an attribute
  - ▶ designed by a schema author

# Defining new Simple Types

- New simple types may be defined from built-in types, adding constraints via "facets"
- Examples:
  - ▶ A string that has a minimum and maximum length
  - ▶ An integer that has minimum and maximum values
  - ▶ A string with an enumerated list of allowed values
  - ▶ A type based on patterns...

# Example: US Postal ("zip") code

- Only these two exact forms are legal:
  - ▶ five digits, or
  - ▶ five digits followed by a dash then four digits
- Examples:
  - ▶ 02155
  - ▶ 02155-2153
- Here's a schema definition for zip codes:

```
<simpleType name="USZipcode"  
            base="string">  
    <pattern value="[0-9]{5}(-[0-9]{4})?"/>  
</simpleType>
```

# Example: Complex Type

<shoeOrder> has

- ▶ a <size> element
- ▶ a ship= attribute

```
<shoeOrder ship='1999-05-21'>  
  <size>9</size>  
</shoeOrder>
```

Here is a schema definition for <shoeOrder>:

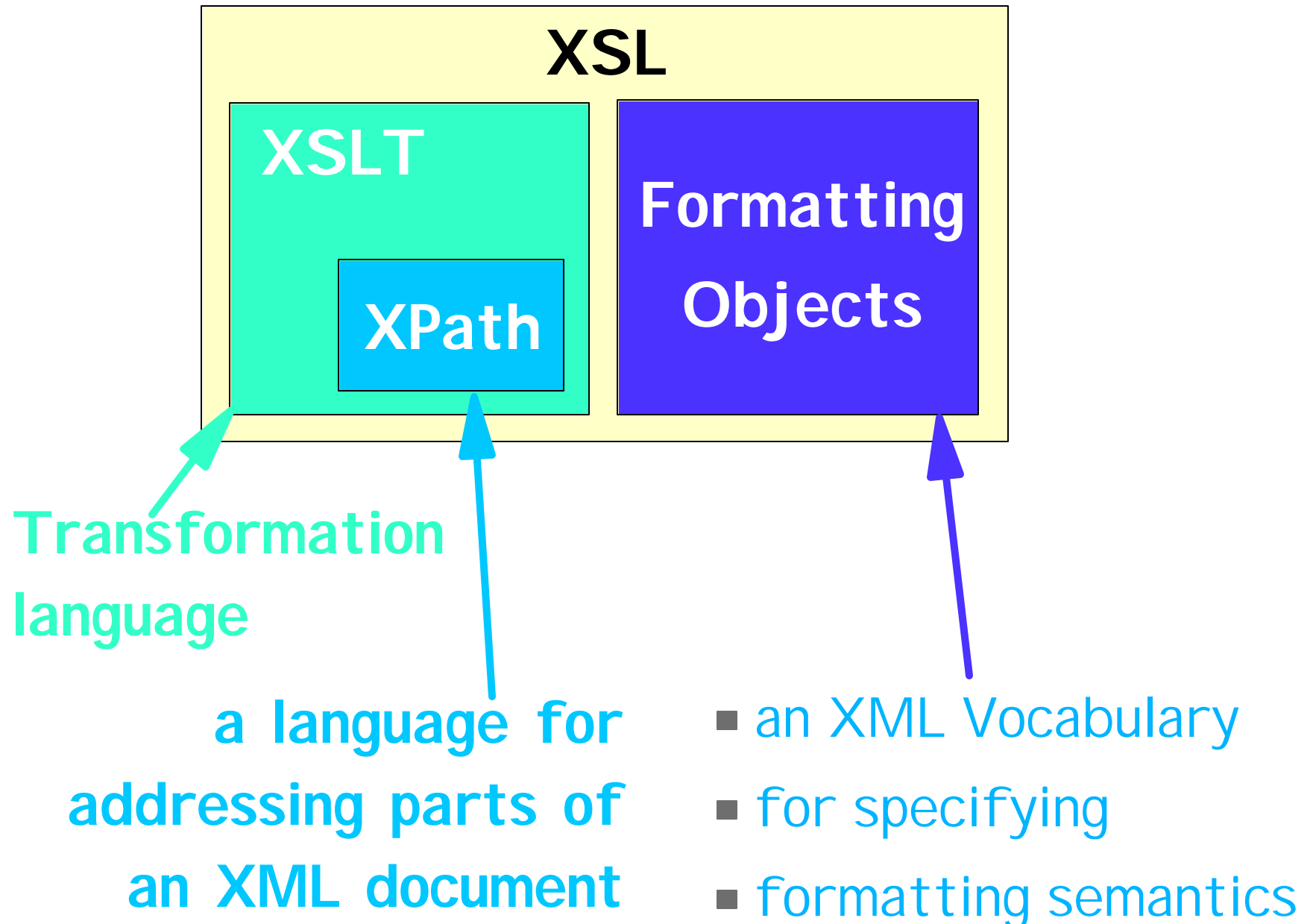
```
<complexType name='ShoeOrder'>  
  <element name='size' />  
  <complexType content='empty'>  
    <attribute name='sz' type='integer' />  
  </complexType>  
  <attribute name='ship' type='date' />  
</complexType>
```

# XSL: Extensible Stylesheet Language

# <term>**XSL**</term>

- **eXtensible Stylesheet Language** is a W3C-defined language for expressing stylesheets and transformations
- An XSL processor provides a mechanism for transforming and formatting XML data, either at the client or on the server
- **Typical uses:**
  - ▶ rendering XML to HTML or PDF
  - ▶ XML vocabulary conversion

# XSL: Three parts



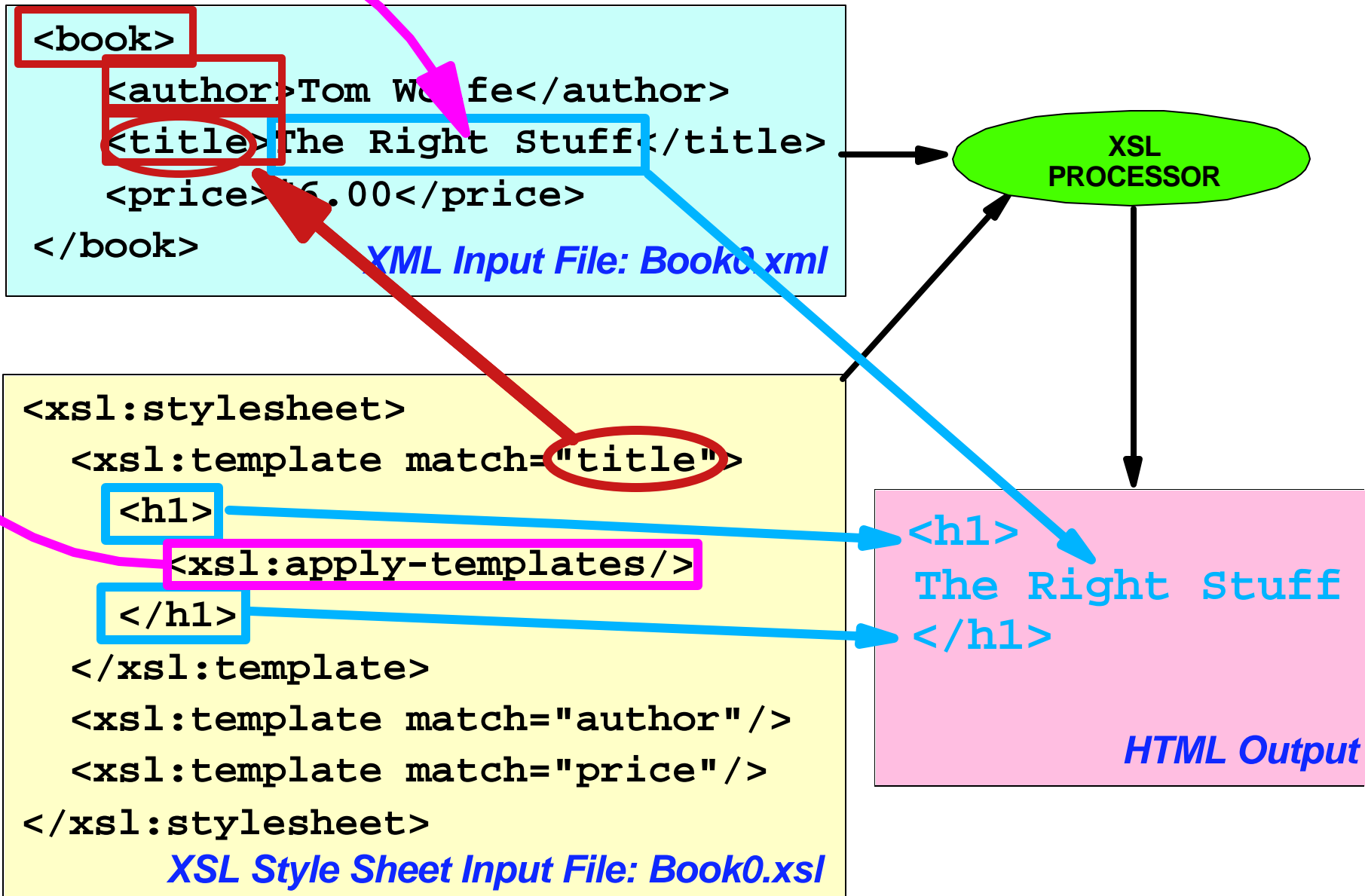
# <term>Transformation</term>

- XML is pointless without transformation!
- "Styling":
  - ▶ prepare data for presentation
  - ▶ e.g. render in html
    - ➔ really a transformation from one XML form to another (HTML)
- "Transformation":
  - ▶ convert from one XML form to another
  - ▶ e.g. vocabulary translation

# XSLT: A Simple Example

(...maybe a little too simple!)

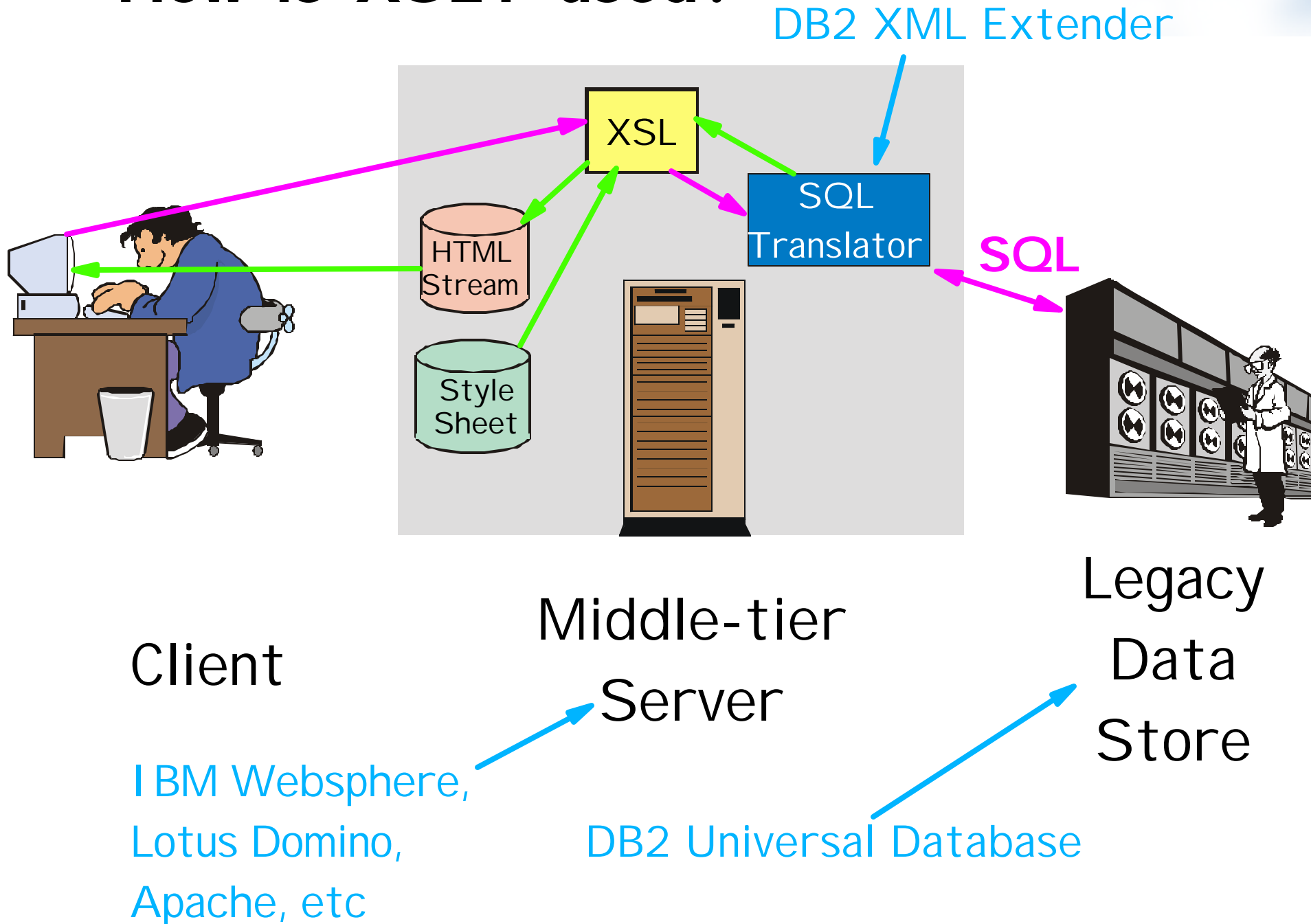
Problem: Display the title (only) in html as <h1> headline



# A slightly larger example

- Transforming an XML file into an HTML table for display in a web browser using XSLT ...
  
- Demos
  - ▶ Embedding an XSL directive in an XML file.
    - ➔ t1.xml / t1.xsl
  
  - ▶ Using a stand-alone style sheet engine.
    - ➔ prime-ministers.xml / pm-html-table2.xsl

# How is XSLT used?



# Summing up XSL Programming

- **XSLT is a language for transformations**
  - ▶ looks for matches to a template
  - ▶ changes data "by example"
  - ▶ styling: rendering to HTML or PDF
  - ▶ transformation: vocabulary translation
- **XPath specifies desired nodes**
  - ▶ works like a pathname of a file
  - ▶ can be used to describe a query
- **Formatting Objects**
  - ▶ define presentation
  - ▶ can render international languages

# What are Web Services?

- The next evolution of e-business
  - ▶ publishing of business functions to the Web
  - ▶ universal access to these functions
- A natural extension to the client-server model
  - ▶ transaction model for e-business
  - ▶ layered services: server can also be a client of services
- e-business is driving the merging of Web, IT & Object technologies to form the foundation for Web Services

# Examples of Web Services

- **Business Information with rich content**
  - ▶ weather reports
  - ▶ news feed
  - ▶ airline schedules
  - ▶ stock quotes
  - ▶ credit check
  - ▶ credit card validation
  - ▶ request for quote
  - ▶ auctions
  
- **Transactional Web Services for B2B or B2C**
  - ▶ airline reservations
  - ▶ rental car agreements
  - ▶ supply chain management
  - ▶ purchase order
  
- **Business Process Externalization**
  - ▶ business linkages at a workflow level
  - ▶ allows complete integration at a process level

# Web Service Components

## ■ Service Provider

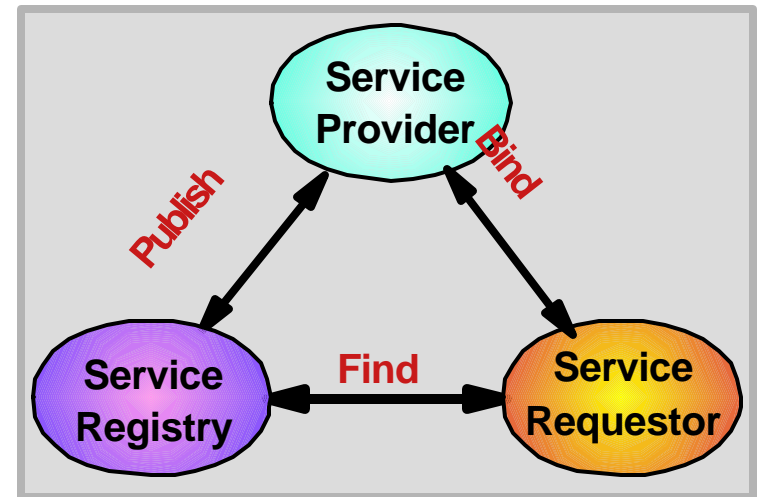
- ▶ provides e-business services
- ▶ **PUBLISHES** availability of these services through a registry

## ■ Service Registry

- ▶ provides support for publishing and locating services
- ▶ like telephone yellow pages

## ■ Service Requestor

- ▶ **FINDS** required services via the Service Registry
- ▶ **BINDS** to services via Service Provider



# Web Services:

## Base Technologies

- **SOAP - Simple Object Access Protocol**
  - ▶ an XML protocol to invoke a method on a server to execute requested operation and get a response in XML
  - ▶ object-oriented programming on web-based objects
    - ➔ but with multiple return values
  - ▶ request message is sent by service requestor
  - ▶ response message is sent by service provider
- **UDDI - Universal Description, Discovery, Integration**
  - ▶ UDDI servers act as a directory of available services and service providers
  - ▶ SOAP can be used to query UDDI for services
- **WSDL - Web Services Description Language**
  - ▶ an XML vocabulary to describe service interfaces

# In Closing...

- Open Standards (like XML) are key enablers of e-business.
- Open Standards and Open Source are not the same thing but they complement each other.
- We are moving towards a Web of services
  - ▶ SOAP provides service-oriented architecture for server-to-server and server-to-device communication
  - ▶ UDDI provides directory for services
- IBM is here to work with you

# Reference Materials

## XML Standards

### Work and Venues

# A Historical Note ...



# W3C XML technologies

## ■ "Recommended" by W3C:

- XML Specification 1.0: syntax, DTDs
- DOM Specification 1.0 & 2.0: Document Model
- XSLT Specification 1.0: transforming XML
- XPath Specification 1.0: queries, addressing XML docs
- XHTML Specification 1.0: HTML in XML form
- XML Schema
- XPointer, XLink
- SOAP 1.2
- DOM 3.0

## ■ Other "standards":

- SAX 2.0 (defacto standard, not from W3C)

# XML 1.0 Specification

- Originally published: February 1998
- In about 35 pages:
  - ▶ XML syntax details
  - ▶ Document Type Definition (DTD)
- XML 1.0 Specification, Second & Third Editions
  - ▶ errata applied to original spec, **not** a new version
  - ▶ replaces Feb 1998 edition, about 55 pages
  - ▶ <http://www.w3.org/TR/2000/REC-xml-20001006>
- Supplementary specs:
  - ▶ Namespaces in XML (January, 1999)
  - ▶ Stylesheet linking (June, 1999)

# XML Schema Specs

- A greatly improved vocabulary definition language
  - ▶ alternative syntax to DTD
  - ▶ XML syntax for defining XML grammars
  - ▶ rich type support
- W3C Working docs: <http://www.w3.org/XML/Schema>
  - ▶ XML Schema Part 0: Primer
  - ▶ XML Schema Part 1: Structures
  - ▶ XML Schema Part 2: Datatypes
- No a W3C Recommendation (May 2nd 2001)

# DOM Specifications

- Models a tree representation of an XML document
  - ▶ tree is created as a result of parsing a document
  - ▶ supports both XML and HTML
- A language-independent object definition and API
  - ▶ Bindings for Java in Appendix
- DOM 1.0 W3C Recommendation: October, 1998
  - ▶ spec: <http://www.w3.org/TR/REC-DOM-Level-1/>
- DOM 2.0 W3C Recommendation: November, 2000
  - ▶ new methods, types, interfaces
  - ▶ traversals, namespaces, event model, stylesheets
- DOM 3.0 W3C Recommendation
  - ▶ Further extends DOM capabilities

# SAX 2.0 Specification

- A de-facto "standard" by Dave Megginson
  - not from W3C
- A free API for event-based XML parsing
  - ▶ instead of getting a complete DOM tree, you get notifications of the arrival of each piece
  - ▶ essential when parsing very large documents
- Available for Java, C++, COM, Perl, Python
- Version 1.0 published May, 1998
- Version 2.0 published May, 2000
- SAX 2.0 support is available in Xerces parser
- see <http://www.megginson.com/SAX/index.html>

# XSLT 1.0 Specification

- A transformation language for XML documents
  - ▶ styling (rendering to visual form, like HTML)
  - ▶ transformation (vocabulary translation)
  - ▶ can emit XML, HTML, even non-XML formats
- XSLT documents are well-formed XML
- W3C Recommendation: November, 1999
  - ▶ spec: <http://www.w3.org/TR/xslt>

# XSL Formatting Objects

- **Layout-oriented XML vocabulary**
  - ▶ rich representation of documents for printing, various device screens, etc
  - ▶ usually created as output of XSLT using an appropriate stylesheet
- **XSL Specification defines FO's, refers to XSLT**
  - ▶ see <http://www.w3.org/Style/XSL>
- **FOP open source FO processor implementation (creates PDF) available at [xml.apache.org](http://xml.apache.org)**

# XPath 1.0 Specification

- Language for addressing parts of an XML document
  - ▶ used by XSLT and XPointer
  - ▶ basic facilities for manipulation of strings, numbers and booleans
  - ▶ can be used as simple query language
  - ▶ compact, non-XML syntax for use in URIs
- W3C Recommendation: November, 1999
  - ▶ see <http://www.w3.org/TR/xpath>

# XLink Specification

- XML elements for links between documents
  - ▶ simple links similar to HTML hypertext links
  - ▶ supports more sophisticated links
- W3C Recommendation: June, 2001
  - ▶ see <http://www.w3.org/XML/Linking>

# XPointer Specification

- **Fragment identifier for URI references that locates XML resources**
  - ▶ based on XPath
  - ▶ allows for examination of internal document structure and choice based on content
  - ▶ address points and ranges as well as whole nodes
  - ▶ locate information by string matching
- **W3C Recommendation: June, 2001**
  - ▶ see <http://www.w3.org/XML/Linking>

# XHTML 1.0 Specification

- Reformulation of HTML 4.01 as XML
  - ▶ documents must be "well-formed" XML
  - ▶ elements and attributes are lower-case only
  - ▶ for non-empty elements, end tags are required
  - ▶ empty elements (`<br />`) allowed
  - ▶ attribute values must always be quoted
  - ▶ no attribute "minimization"
- W3C Recommendation: January 2000
  - ▶ spec: <http://www.w3.org/TR/xhtml1/>

# Questions?

[ibm.com /alphaworks](http://ibm.com/alphaworks)

site for free emerging tools and technologies from IBM

[ibm.com/developer/xml](http://ibm.com/developer/xml)

XML Zone on developerWorks - resources for customers and developers on the use of XML

[xml.apache.org](http://xml.apache.org)

open source XML tools from Apache Software Foundation

[www.w3.org](http://www.w3.org)

XML base technical standards

[xml.org](http://xml.org)

XML standard vocabularies repository

[xml.org/xmlorg\\_news/index.shtml](http://xml.org/xmlorg_news/index.shtml)

new and news (the Cover pages)

[ebxml.org](http://ebxml.org)

electronic business in XML initiative