

Electronic Data Interchange (EDI)

- Standard of formats for common business documents
- Provides for the exchanging of documents
- Used in point-to-point exchanges
- Now can be used over the internet



XML & Data Exchange

- eXtensible Markup Language
 - Offers advantages over other methods
 - Developed by W3C
- Not dependant on any hardware/software
- Does not do anything
- Importance: Enables data **sharing**
- Can be used on the Web



XML

- Allows interaction between:
 - independent computer systems
 - different operating systems
- Allows independent systems to interact with one another
 - Defining data (structure, syntax & type)
 - Validating, interpreting & processing data



XML

- Applications may be written in any programming language
- Integration of applications
- Organization/Industry can define tags for their specific data
 - examples:
 - Supplier name
 - Contract number



Using XML in Business

- XML can improve efficiencies
 - Structure defined in an XML schema document
 - Documents created in XML
 - Information sent is validated against schema
 - Schema may be published on the Web
 - Granting access to all businesses to allow them to validate documents received



Examples of XML Industry Standards

- Financial Reporting
 - XBRL, www.xbrl.org
- General Ledgers
 - XBRL GL, xml.coverpages.org/ni2001-06-21-b.html
- eContracts Business Narrative
 - BNML, xml.coverpages.org/ni2005-08-11-a.html
- Insurance
 - ACORD, www.acord.org
- Human Resources
 - HR-XML, www.hr-xml.org

XML site - <http://www.xml.org>

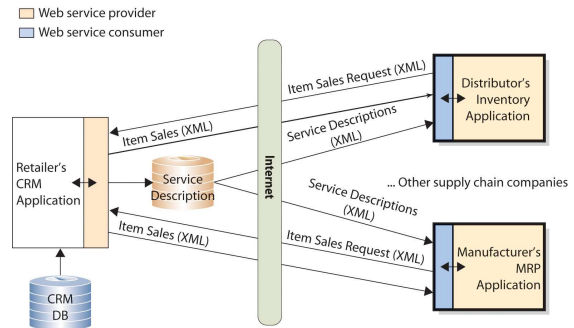


XML Web Services

- Web services
 - Most software vendors support XML
 - Standards that facilitate distributed computing
 - Use the Internet
 - Tool for application interaction
 - Allows programs to access each other remotely
 - No need to develop proprietary system
 - Data transmitted in XML documents
 - XML schemas defined
 - Automatic validation



Web Services



XML

- Simple "easy" format
- XML
 - Document Type Definition (DTD)
 - XML Schema
 - Document (data)
- Text file with tags
 - similar format to HTML
 - No predefined tags



XML Document

- XML tags
 - Define data (structure, syntax & type)
- Metadata (text)
 - data about the data



XML Document

- 1st line, XML declaration (optional)
 - **Version**, Version 1.0, 1998, editions:
 - *2nd - 2000, 3rd - 2004, *4th - 2006
 - * Considered the current versions
 - **Character encoding**, default UTF-8
 - **Dependencies**

`<?XML version="1.0" encoding="UTF-8" Standalone="yes"?>`

Required Syntax



XML Document - Elements

- Tags within the document
- Start & End Tags: `<name>` content `</name>`
- Tags are case sensitive (name \neq NAME)
- Create descriptive tag names (**no spaces**)
- Document should have 1 root element
- Structure defined by the grouping and/or nesting of elements
 - elements can follow other elements
 - elements can be nested within other element



XML Document - Attributes

- Additional data to describe elements
- Name/Value pairs, in the start tag
- Attribute value contained in quotes
- `<name attribute="value">content</name>`



Well Formed XML Document

- Complies to a character set definition
- One & only one root element
- Elements delimited by start & end tag
- Tags nested not overlapping
- Attribute values quoted



XML Example

Employee travel expenses:

Employee:

 Name

 Department

Expense:

 Date

 Item

 Amount

XML Example

Employee travel expenses:

Employee:

 Name: Jane Winston

 Department: Accounts Payable

Expense:

 Date: 2007-01-07

 Item: Airfare

 Amount: \$500

Expense:

 Date: 2007-01-07

 Item: Hotel

 Amount: \$200

Expense:

 Date: 2007-01-07

 Item: Taxi Fare

 Amount: \$50

XML Example

```

<?xml version="1.0"?>
<EmployeeExpenses>
  <EmployeeInfo>
    <Name>Jane Winston</Name>
    <Department>Accounts Payable</Department>
  </EmployeeInfo>
  <ExpenseItem>
    <Date>2007-01-07</Date>
    <Description>Airfare</Description>
    <Amount>500.34</Amount>
  </ExpenseItem>
  <ExpenseItem>
    <Date>2007-01-07</Date>
    <Description>Hotel</Description>
    <Amount>200</Amount>
  </ExpenseItem>
  <ExpenseItem>
    <Date>2007-01-07</Date>
    <Description>Taxi Fare</Description>
    <Amount>50.00</Amount>
  </ExpenseItem>
</EmployeeExpenses>
    
```


XML Example (attributes)

```

<?xml version="1.0"?>
<EmployeeExpenses>
  <EmployeeInfo>
    <Name>Jane Winston</Name>
    <Department>Accounts Payable</Department>
  </EmployeeInfo>
  <ExpenseItem>
    <Date>2007-01-07</Date>
    <Description>Airfare</Description>
    <Amount currency="CDN">500.34</Amount>
  </ExpenseItem>
  <ExpenseItem>
    <Date>2007-01-07</Date>
    <Description>Hotel</Description>
    <Amount currency="CDN">200</Amount>
  </ExpenseItem>
  <ExpenseItem>
    <Date>2007-01-07</Date>
    <Description>Taxi Fare</Description>
    <Amount currency="US">50.00</Amount>
  </ExpenseItem>
</EmployeeExpenses>
    
```


XML Namespace

- XML document may contain Elements from different sources and these Elements may have duplicate names
- Provides method to avoid element names conflicts
- Used in XML schemas



XML Namespace

- Placed in the start tag of an element
- Syntax:
 - xmlns:namespace-prefix="namespaceURI"
- Solve name conflicts with prefix
- "namespaceURI"
 - Namespace Uniform Resource Identifier
 - URI - identifies an Internet Resource
 - Provides a unique name for namespace
 - The most common URI is a URL (web page)



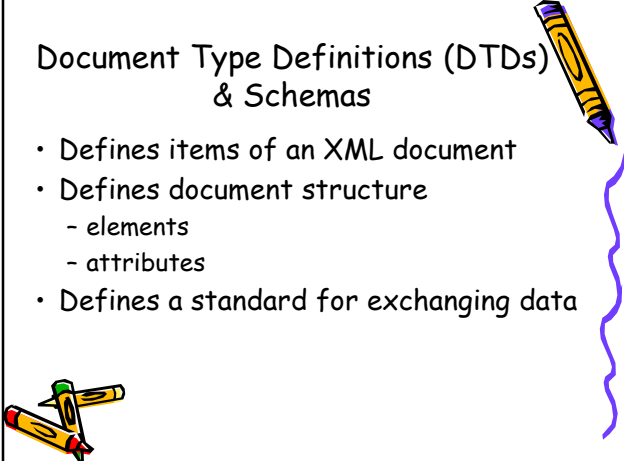
```

<?xml version="1.0"?>
<ns1:EmployeeExpenses xmlns:ns1="http://www.mybusiness.com">
  <ns1:EmployeeInfo>
    <ns1:Name>Jane Winston</ns1:Name>
    <ns1:Department>Accounts Payable</ns1:Department>
  </ns1:EmployeeInfo>
  <ns1:ExpenseItem>
    <ns1:Date>2007-01-07</ns1:Date>
    <ns1:Description>Airfare</ns1:Description>
    <ns1:Amount>500.34</ns1:Amount>
  </ns1:ExpenseItem>
  <ns1:ExpenseItem>
    <ns1:Date>2007-01-07</ns1:Date>
    <ns1:Description>Hotel</ns1:Description>
    <ns1:Amount>200</ns1:Amount>
  </ns1:ExpenseItem>
  <ns1:ExpenseItem>
    <ns1:Date>2007-01-07</ns1:Date>
    <ns1:Description>Taxi Fare</ns1:Description>
    <ns1:Amount>50.00</ns1:Amount>
  </ns1:ExpenseItem>
</ns1:EmployeeExpenses>
    
```

**XML
Namespace
Example**

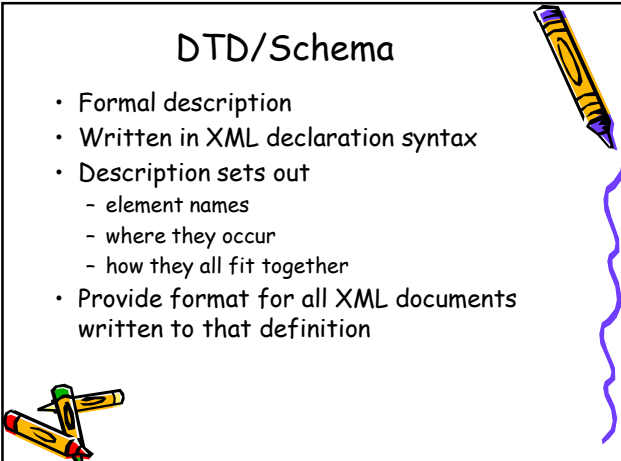
Document Type Definitions (DTDs) & Schemas

- Defines items of an XML document
- Defines document structure
 - elements
 - attributes
- Defines a standard for exchanging data



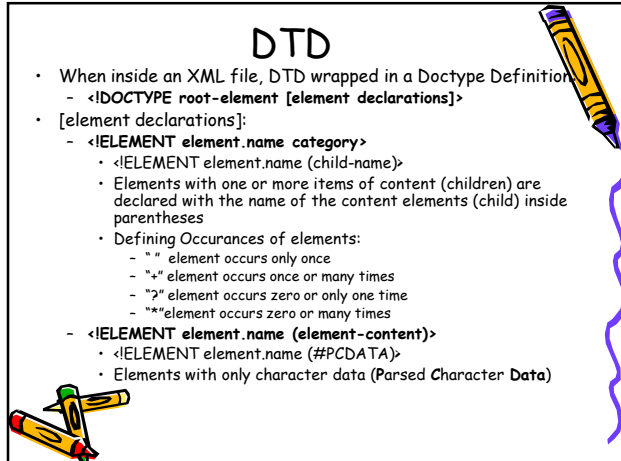
DTD/Schema

- Formal description
- Written in XML declaration syntax
- Description sets out
 - element names
 - where they occur
 - how they all fit together
- Provide format for all XML documents written to that definition



DTD

- When inside an XML file, DTD wrapped in a Doctype Definition
 - <!DOCTYPE root-element [element declarations]>
- [element declarations]:
 - <!ELEMENT element.name category>
 - <!ELEMENT element.name (child-name)>
 - Elements with one or more items of content (children) are declared with the name of the content elements (child) inside parentheses
 - Defining Occurances of elements:
 - "" element occurs only once
 - "+" element occurs once or many times
 - "?" element occurs zero or only one time
 - "*" element occurs zero or many times
 - <!ELEMENT element.name (element-content)>
 - <!ELEMENT element.name (#PCDATA)>
 - Elements with only character data (Parsed Character Data)



XML Example

Recipe

Name - JELL-O

Ingredients

- one box of JELL-O (gelatin)
- 250 ml boiling water
- 250 ml cold water

Instructions

- Add boiling water to gelatin. Stir until completely dissolved. Add cold water. Chill until set



XML Example

Recipe

Name

Ingredient(s)

Item, Quantity

Instructions



DTD Example

```
<!Doctype Recipe[
<!ELEMENT
  Recipe (Name, Ingredients, Instructions)>
<!ELEMENT Name (#PCDATA)>
  <!ELEMENT Ingredients (Ingredient)+>
<!ELEMENT Ingredient (Qty, Item)>
<!ELEMENT Qty (#PCDATA)>
<!ELEMENT Item (#PCDATA)>
<!ELEMENT Instructions (#PCDATA)> ]>
```



Why?

- XML Schemas:
 - Successors to DTDs
 - Standard set by the World Wide Web Consortium (W3C)
 - Expected to soon replace DTDs
 - are extensible to future additions
 - richer and more powerful than DTDs
 - written in XML
 - support data types
 - support namespaces



XML Schema

- Defines
 - elements that can appear in a document
 - attributes that can appear in a document
 - data types for elements and attributes
 - default & fixed values for elements & attributes



XML Schema

- XML Schema Definition (XSD)
 - description of the structure of an XML document
- Schema definitions are XML documents



XML Schema Elements

- Root element (one per schema)
- Declaration elements
- Defining elements
- Defining content models



Schema Elements

- Root Element includes
 - a declaration of the namespaces (xmlns) used in this definition
 - the version of the schema
 - the language used in the schema
 - plus many other attributes

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```



Schema Elements

- `xsd:element`
 - Declares an element in the document.
- Attributes of an element include:
 - `name` - the name of the element
 - `minOccurs`
 - minimum number of occurrences
 - default is 1
 - for optional use "0"
 - `maxOccurs`
 - maximum number of occurrences
 - default is 1
 - for unlimited use "unbounded"



Schema Elements

- Defining elements
 - "complexType"
 - Used to group elements
 - `xsd:complexType`
- Defining content models
 - "sequence"
 - Defines the sequence for elements in a group
 - `xsd:sequence`



XML Example

Recipe
 Name
 Ingredient(s)
 Item, Quantity
 Instructions



```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Recipe">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Name"></xsd:element>
        <xsd:element maxOccurs="unbounded" name="Ingredient">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="Item"></element>
              <xsd:element name="Qty"></element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="Instructions"></xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

XML Example

Recipe

Name - JELL-O

Ingredients

- one box of JELL-O (gelatin)
- 250 ml boiling water
- 250 ml cold water

Instructions

- Add boiling water to gelatin. Stir until completely dissolved. Add cold water. Chill until set



XML Document

```
<?xml version="1.0"?>  
<Recipe>  
  <Name>Jell-O</Name>  
  <Ingredient>  
    <Item>gelatin</Item> <Qty>1</Qty>  
  </Ingredient>  
  <Ingredient>  
    <Item>boiling water</Item> <Qty>250</Qty>  
  </Ingredient>  
  <Ingredient>  
    <Item>cold water</Item> <Qty>250</Qty>  
  </Ingredient>  
  <Instructions> Add boiling water to gelatin. Stir until  
  completely dissolved. Add cold water. Chill until set.  
  </Instructions>  
</Recipe>
```

XML Example

Employees at Western

Employee: Fred
Office: MC 110
Phone: 555-1256
Employee: Sue
Phone: 555-0090
Employee: Dave
Office: SSC 6712



Data Structure:

Employees

Employee (many)
Name
Office (optional)
Phone (optional)



```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Employees">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element maxOccurs="unbounded" name="Employee">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="Name"/>
              <xsd:element minOccurs="0" name="Office"/>
              <xsd:element minOccurs="0" name="Phone"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

XML Schema

XML Example

Employees at Western
 Employee: Fred
 Office: MC 110
 Phone: 555-1256
 Employee: Sue
 Phone: 555-0090
 Employee: Dave
 Office: SSC 6712



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Employees>
  <Employee>
    <Name>Fred</Name>
    <Office>MC 110</Office>
    <Phone>555-1256</Phone>
  </Employee>
  <Employee>
    <Name>Sue</Name>
    <Phone>555-0090</Phone>
  </Employee>
  <Employee>
    <Name>Dave</Name>
    <Office>SSC 6712</Office>
  </Employee>
</Employees>
```

XML DOCUMENT

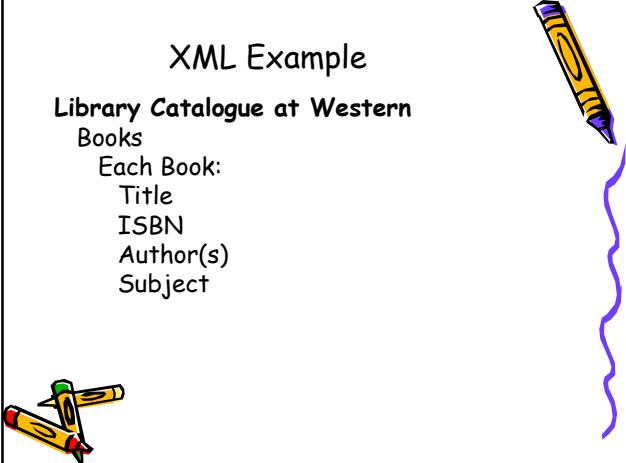
Library Catalogue at Western

Books
 Book:
 Title: Experiencing MIS Canadian Edition
 ISBN: 0132396203
 Author: D. M. Kroenke, A. Gemino, P. Tingling
 Subject: Management Information Systems
 Book:
 Title: XML in A Nutshell
 ISBN: 0596002920
 Authors: E. R. Harold, W. S. Means
 Subject: XML



XML Example

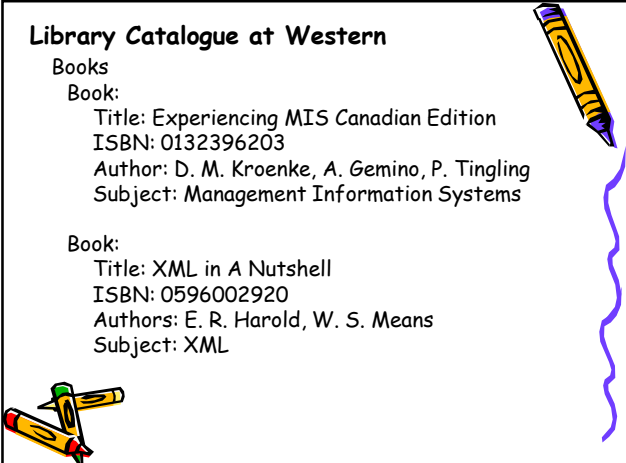
Library Catalogue at Western
Books
 Each Book:
 Title
 ISBN
 Author(s)
 Subject



```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="uwoLibrary">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Books">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element maxOccurs="unbounded" name="Book">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="Title"/></xsd:element>
                    <xsd:element name="ISBN"/></xsd:element>
                    <xsd:element maxOccurs="unbounded"
                      name="Author"/></xsd:element>
                    <xsd:element name="Subject"/></xsd:element>
                  </xsd:sequence> </xsd:complexType>
                </xsd:element>
              </xsd:sequence> </xsd:complexType>
            </xsd:element>
          </xsd:sequence> </xsd:complexType>
        </xsd:element>
      </xsd:sequence> </xsd:complexType>
    </xsd:element>
  </xsd:schema>
```

Library Catalogue at Western
Books
 Book:
 Title: Experiencing MIS Canadian Edition
 ISBN: 0132396203
 Author: D. M. Kroenke, A. Gemino, P. Tingling
 Subject: Management Information Systems

Book:
 Title: XML in A Nutshell
 ISBN: 0596002920
 Authors: E. R. Harold, W. S. Means
 Subject: XML



```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<uwoLibrary>
  <Books>
    <Book>
      <Title>Experiencing MIS</Title>
      <ISBN>0132337770</ISBN>
      <Author>D. M. Kroenke</Author>
      <Author>A. Gemino</Author>
      <Author>P. Tingling</Author>
      <Subject>Management Information Systems</Subject>
    </Book>
    <Book>
      <Title>XML in A Nutshell</Title>
      <ISBN>0596002920</ISBN>
      <Author>E. R. Harold</Author>
      <Author>W. S. Means</Author>
      <Subject>XML</Subject>
    </Book>
  </Books>
</uwoLibrary>
```

XML Document