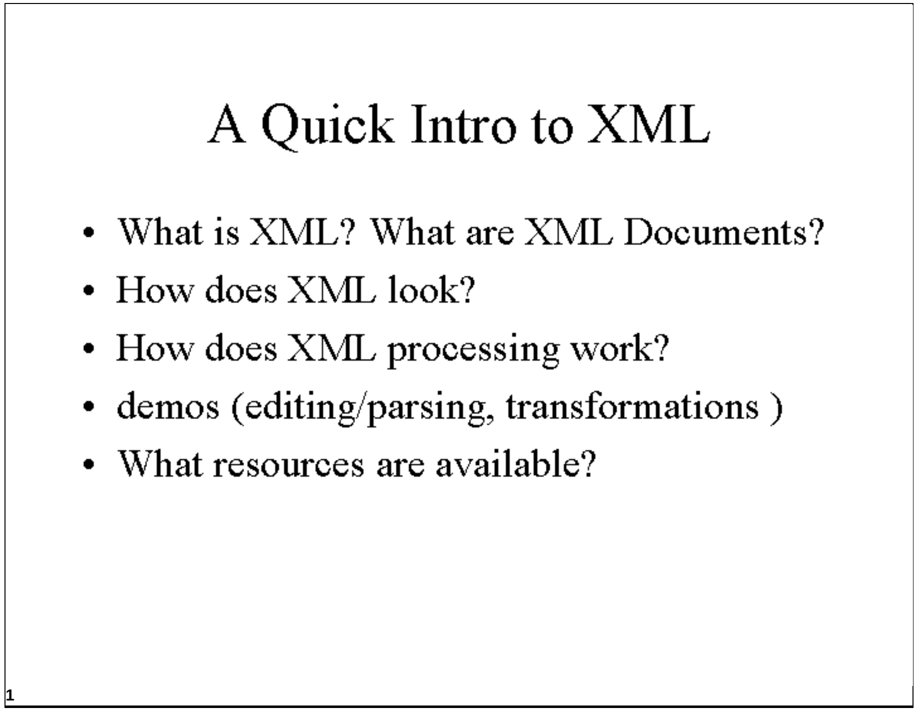
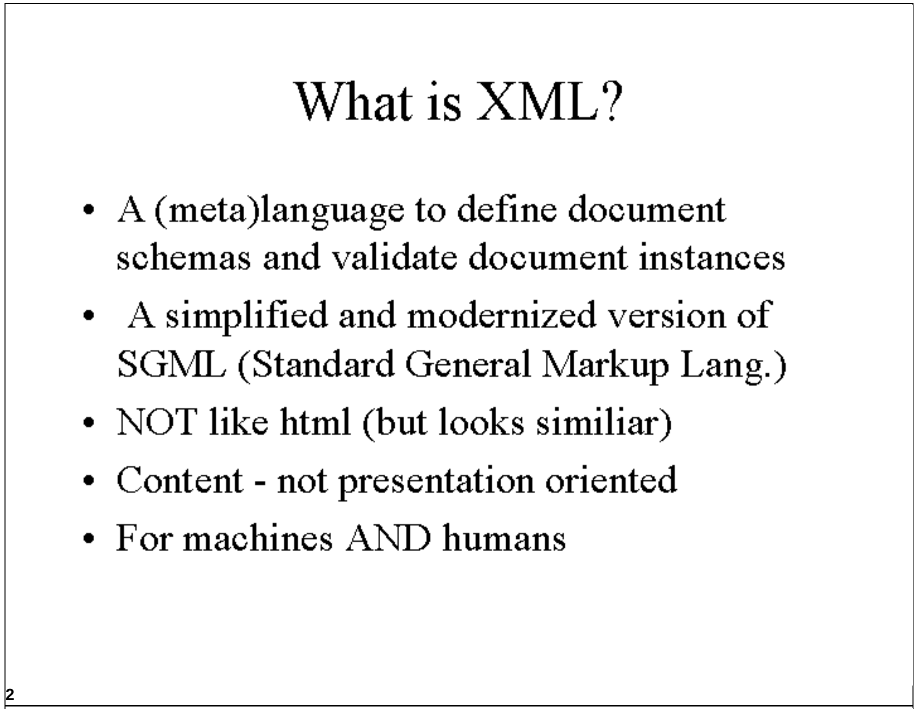
XML Basics



A Quick Intro to XML

- What is XML? What are XML Documents?
- How does XML look?
- How does XML processing work?
- demos (editing/parsing, transformations)
- What resources are available?

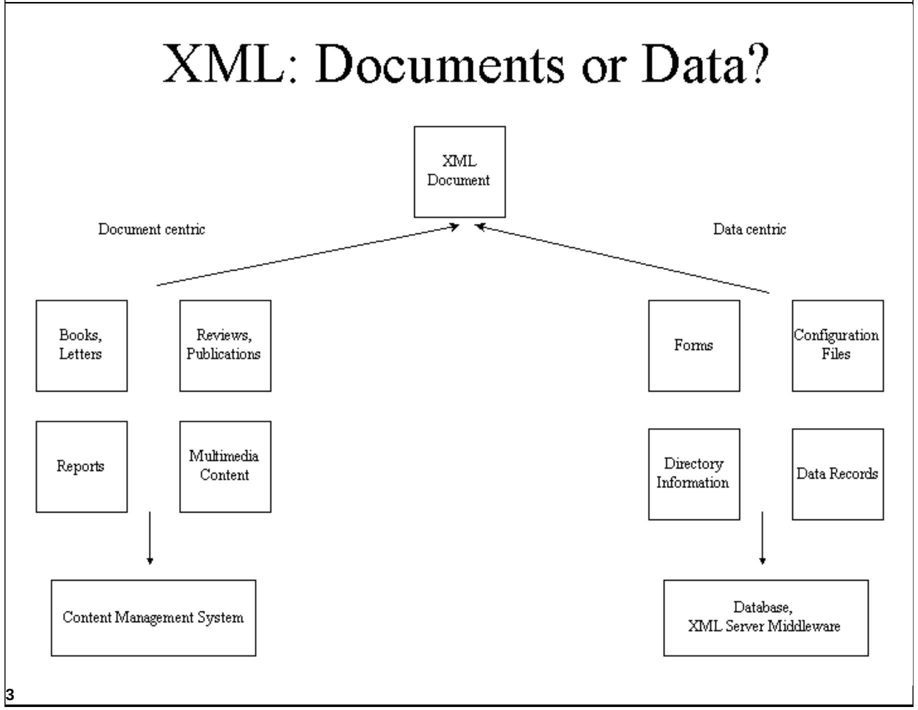
1



What is XML?

- A (meta)language to define document schemas and validate document instances
- A simplified and modernized version of SGML (Standard General Markup Lang.)
- NOT like html (but looks similiar)
- Content - not presentation oriented
- For machines AND humans

2



XML: Documents or Data?

XML Document

Document centric

Data centric

Books, Letters

Reviews, Publications

Reports

Multimedia Content

Content Management System

Forms

Configuration Files

Directory Information

Data Records

Database, XML Server Middleware

3

Standards

- **XML**: Extensible Markup Language V1.0
- **XSL**: Extended Stylesheet Language
- **XPath**: A way to address things
- **Xlink**: A way to link things
- **RDF**: a way to express semantics
- **Sax**: Parser interface
- **DOM** (Document Object Model)
- **XSD**: XML Schema Definition Language
- **XHTML**: the new HTML in XML syntax

Advantages of XML information

- human and machine readable
- no fixed tag set, any schema is possible
- Content centric and self-describing
- well-formed or valid
- tool independence
- A data definition in XML is descriptive code AND its documentation – immediately usable by clients

Technical Terms 1

- **DTD**: Document Type Definition. A way to specify the structure and content of documents
- **Instance**: a concrete document
- **DOM** (Document Object Model): A way to programmatically access XML elements and attributes

Technical Terms 2

```
<car>volvo</car>
```

```
<car color=„red“
```

```
<car>&bmw;</car>
```

```
<? handle separately>
```

- **Elements**: The basic structures of a document
- **Attributes**: Meta-information about elements
- **Entities**: Kind of “macros” or includes
- **Process. Instructions**: hints for applications

How does XML look?

An XML instance

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE teilnahmeb SYSTEM "teilnahmeb.dtd" [
        <!ENTITY lt "&#60;">
        <!ENTITY gt "&#62;">
        <!ENTITY amp "&#38;">
        <!ENTITY class "XML BASIC">
]>
<teilnahmeb Autor=„Kriha“>
        <nachweis>Frau <teiln>Nina Schwarz</teiln> hat am
        <datum>22.9.98</datum> am Kurs <kursname>&class;</kursname>
teilgenommen.
<?Pub Caret?>
        </nachweis>
        <kursinfo>
                <absatz>Introduction to XML </absatz>
        </kursinfo>
        <adresse>
                <name>Walter Kriha</name>
                <strasse>Schwarzwaldstr.7g</strasse>
        </adresse>
</teilnahmeb>
```

An XML DTD

```
<!ELEMENT teilnahmeb (nachweis,kursinfo,adresse,kommentar*)>

<!ATTLIST teilnahmeb Autor #CDATA>

<!ELEMENT kommentar (#PCDATA)>
<!ELEMENT nachweis (#PCDATA | teiln | datum | kursname )*>
<!ELEMENT kursinfo (absatz+)>
<!ELEMENT adresse (name,strasse,ort)>
<!ELEMENT teiln (#PCDATA)>
<!ELEMENT datum (#PCDATA)>
<!ELEMENT kursname (#PCDATA)>
<!ELEMENT absatz (#PCDATA | hervorh | fussnote)*>
<!ELEMENT hervorh (#PCDATA)>
<!ELEMENT fussnote (absatz+)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT strasse (#PCDATA)>
<!ELEMENT ort (#PCDATA)>
```

Formal or Domain Markup

FORMAL

```
<message>
 <command target="accounting" id="1">
   <process>update</process>
   <object class="GlAccount" oid="12345"
     version="1">
     <context>
       <owner class="ChartOfAccounts"
     oid="47"/>
     </context>
     <property name="name">Bank
     Account</property>
     <property name="type">Asset</property>
     <object name="balance" class="Money">
        <property
     name="currency">USD</property>
        <property name="amount"
     type="float">15000.00</property>
     </object>
   </object>
 </command>
</message>
```

DOMAIN SPECIFIC

```
<message>
 <update target="accounting" id="1">
  <GlAccount oid="12345" version="1">
   <ChartOfAccounts oid="47"/>
   <GlAccount.name>Bank
    Account</GlAccount.name>
   <GlAccount.type>Asset</GlAccount.type>
   <Balance>
    <Currency>USD</Currency>
    <Amount>15000.00</Amount>
   </Balance>
  </GlAccount>
 </update>
</message>
```

An XSL style sheet

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl" xmlns:html="http://www.w3.org/TR/REC-
html40/" result-ns="html">
        <!– Root Element -->
        <xsl:template match="/">
        <HTML>    <HEAD>
                <xsl:for-each select="title">
                        <title>
                                <xsl:process-children/>
                        </title>
                </xsl:for-each>
                </HEAD>
                <body>
                        <xsl:process-children/>
                </body>
        </HTML>
        </xsl:template>
        <xsl:template match="account">
                <h1>
                        <xsl:process-children/>
                </h1>
        </xsl:template>
        <xsl:template match="firstname">
                <p>
                        <xsl:process-children/>
                ---- more -------
```

The bold parts match elements from the xml input document

Xpointer/Xlink elements

```
<?xml ver="1.0"?>
<!DOCTYPE doc PUBLIC "-//Masatomo Goto//DTD XLink sample document//EN" [
<?STYLESHEET href="..\styles\sample_scroll.dsl" type="text/dsssl" ?>

]>
<doc>
<title>XLink and XPointer: how do I address and link

<group steps="2">
  <groupdoc href="dsssloverview.xml"
  <groupdoc href="readme
</group>
<xlink>
  <locator role="Hubdocument Title" href="myhub.xml#root().child( 1, title)"/>
  <locator role="Overview Title" href="readme.xml#root().child( 1, title)"/>
  <locator role="DSSSL Spec Reference" href="readme.xml#root().child( 1, chapter).(2,
section).( 2, p)"/>
  <locator show="new" role="DSSSL Spec" href="dsssloverview.xml#root().child( 1,
title)"/>
</xlink>
<!--
show="new"
-->

<xlink>
  <locator role="toc" href="#root().child( 2, chapter).(1, section).(4, p)"/>
  <locator role="cont." href="readme.xml#child( 1, chapter).(4, section).( 1, title)"/>
</xlink>

------ more ---------------
```

This Xlink connects 4 locations in 3 different documents. Every locator serves a different role.

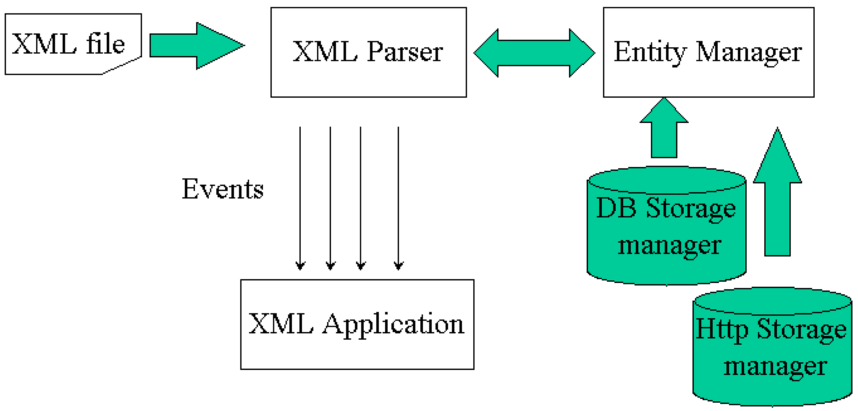
How does XML processing work?

Parser Input: Instance + DTD

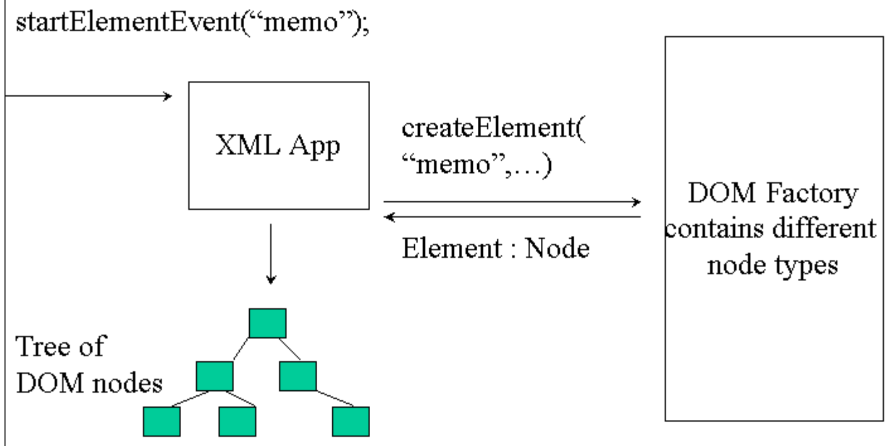
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE teilnahmeb SYSTEM "teilnahmeb.dtd" [
          <!ENTITY lt "&#60;">
          <!ENTITY gt "&#62;">
          <!ENTITY amp "&#38;">
          <!ENTITY class "XML BASIC">
]>
<teilnahmeb>
          <nachweis>Frau <teiln>Nina Schwarz</teiln> hat am <datum>22.9.98</datum> am Kurs
<kursname>&class;</kursname> teilgenommen.<?Pub Caret?>
          </nachweis>
          <kursinfo>
                    <absatz>Introduction to XML </absatz>
          </kursinfo>
          <adresse>
                    <name>Walter Kriha</name>
                    <strasse>Schwarzwaldstr.7g</strasse>
          </adresse>
</teilnahmeb>
```

```
<!ELEMENT teilnahmeb (nachweis,kursinfo,adresse,kommentar*)>
<!ELEMENT kommentar (#PCDATA)>
<!ELEMENT nachweis (#PCDATA | teiln | datum | kursname )*>
<!ELEMENT kursinfo (absatz+)>
<!ELEMENT adresse (name,strasse,ort)>
<!ELEMENT teiln (#PCDATA)>
<!ELEMENT datum (#PCDATA)>
<!ELEMENT kursname (#PCDATA)>
<!ELEMENT absatz (#PCDATA | hervorh | fussnote)*>
<!ELEMENT hervorh (#PCDATA)>
<!ELEMENT fussnote (absatz+)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT strasse (#PCDATA)>
<!ELEMENT ort (#PCDATA)>
```

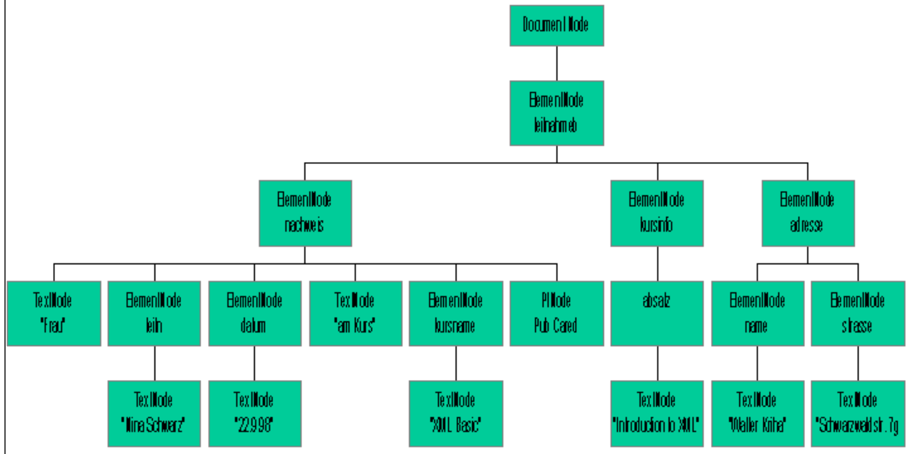
XML File to XML Application



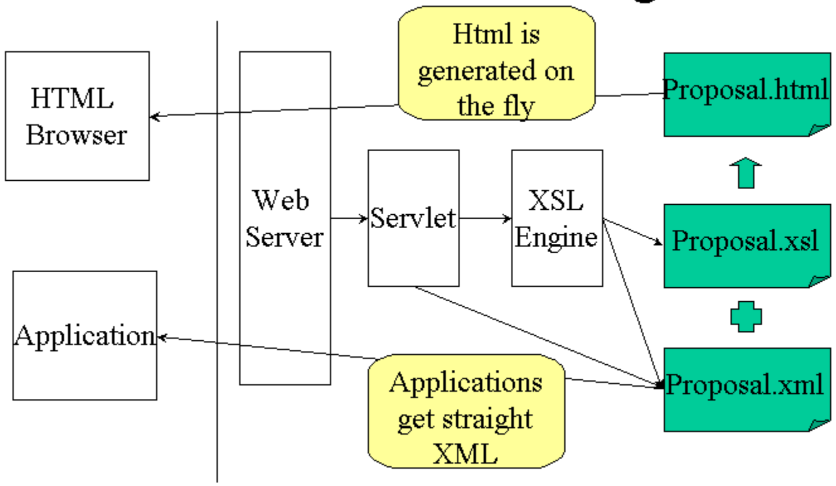
XML Application to DOM


startElementEvent("memo");
XML App
createElement("memo",...)
Element : Node
DOM Factory contains different node types
Tree of DOM nodes

teilnahmeb as DOM tree



Server Side Processing



Demo1: Editing/Parsing

- Parser: NSGMLS parser from James Clark
- Editor: XEmacs, XMLSpy, wordpad, vi
- Objective: What does it mean to VALIDATE a document?
- Resources: teilnahmeb.dtd (a DTD), teilnahmeb.xml (a valid instance), teilnahmeb1.xml (an invalid instance)

From DTDs to Schemas

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
          <xs:element name="absatz">
                    <xs:complexType mixed="true">
                              <xs:choice minOccurs="0" maxOccurs="unbounded">
                                        <xs:element ref="hervorh"/>
                                        <xs:element ref="fussnote"/>
                              </xs:choice>
                    </xs:complexType>
          </xs:element>
          <xs:element name="adresse">
                    <xs:complexType>
                              <xs:sequence>
                                        <xs:element ref="name"/>
                                        <xs:element ref="strasse"/>
                                        <xs:element ref="ort"/>
                              </xs:sequence>
                    </xs:complexType>
          </xs:element>
```

```
<!ELEMENT teilnahmeb (nachweis,kursinfo,adresse,kommentar*)>
<!ELEMENT kommentar (#PCDATA)>
<!ELEMENT nachweis (#PCDATA | teiln | datum | kursname )*>
<!ELEMENT kursinfo (absatz+)>
<!ELEMENT adresse (name,strasse,ort)>
<!ELEMENT teiln (#PCDATA)>
<!ELEMENT datum (#PCDATA)>
<!ELEMENT kursname (#PCDATA)>
<!ELEMENT absatz (#PCDATA | hervorh | fussnote)*>
<!ELEMENT hervorh (#PCDATA)>
<!ELEMENT fussnote (absatz+)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT strasse (#PCDATA)>
<!ELEMENT ort (#PCDATA)>
```

Job Description Format: an XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.CIP4.org/JDFSchema_1"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:jdfP="http://www.CIP4.org/JDFSchema_1/JDFParser"
     xmlns:jdf="http://www.CIP4.org/JDFSchema_1"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
     attributeFormDefault="unqualified">
<!--Base Elements from which all other elements are derived -->
<xsd:complexType name="EmptyElement"/>
<xsd:complexType name="Comment_Type" mixed="true">
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:attribute name="Box"
type="jdf:rectangle" use="optional"/>
<xsd:attribute name="Language"
type="xsd:language" use="optional"/>
<xsd:attribute name="Name"
type="xsd:NMTOKEN" default="Description"/>
<xsd:attribute name="Path" type="jdf:path"
use="optional"/>… ....................
```

Please read the introduction to XSD (see resources)

Demo2: XSL transformations

- XSL engine: saxon (java) or mozilla/firefox browser or IE
- Objective: What does it mean to TRANSFORM a document using stylesheets?
- Resources: catalog.xml/xsl, article.xml/xsl
- Results: html versions of those

XML Resources

- Robin Covers page, find everything about SGML/XML at: www.oasis-open.org/cover
- xml-dev: mailing list for XML developers
- Charles Goldfarb, The XML Handbook
- IBM xml tools in java: www.alphaworks.ibm.com
- www.xmlsoftware.com (tools)
- www.xml.com (news)
- http://www.editor.net/intro.htm, explains writing for the Internet

XSL Resources

- http://www.webdevelopersjournal.com/articles/xml_to_html.html a good introduction to xsl conversions by Benoit Marchal (with samples)
- Michael Kay, Professional XML programming (the „bible“)
- frequently asked questions: http://www.dpawson.co.uk/xsl/sect1/sect1.html (excellent and free)

XML Schema (XSD) Resources

- http://www.w3schools.com/schema/ a short introduction to XSD with samples. We will need XSD for the Job Definition Format of the printing industry!