



- There is, as yet no clear and simple explanation of the various "Schema" efforts. These include:
   Various Structural Schema to replace DTDs

  - RDF Schema
  - Datatypes
  - Namespaces
- The RDF specification is about the clearest: "RDF Specification is about the createst. " "RDF Schemas might be contrasted with XML Schemas ... an XML Schema gives specific constraints on the structure of an XML document ... an RDF Schema provides information about the interpretation of the statements given in an RDF data model....RDF uses XML for its interchange encoding ... XML [datatypes] should be the foundation" September 4, 2001

XML Schema

#### Simplification

- The XML Schema efforts extend the SGML Document Type Definition (DTD) functionality
  - The DTD was designed as a means for defining the structural properties of a class of documents.
- Schema provide an alternative form for defining a documents structure. They also:
  - Allow for more precise control of the content in a documentvia the datatype extensions
  - Allow for more than one definition to be applied within a document via the namespaces extension
  - Allow for simplification of parsing engines by defining schema in the form of XML documents

XML Schema

September 4, 2001

#### Definitions

- "Document instance" is a single document, in our case made up of elements that are hierarchically nested and encapsulated by begin and end tags.
- "Document Type Definition" is a description of a class of document instances that can be used to validate a given document.
- "Schema" is a description of a class of document instances, and is itself a document instance, that can be used to validate a given document.
- "Namespace" is a specification or schema. In the case of a specification, the specification defines the document class.
   September 4, 2001 XML Schema

#### Complications

- Schema work has been complicated in that it is directed at a number of different goals.
- It is further complicated by the fact that there have been competing specifications
  - The Document Content Description (DCD) effort was one proposal by Textuality, Microsoft, and IBM
  - http://www.w3.org/TR/NOTE-dcd
    Document Definition Markup Language (DDML) effort was another proposal by the Europeans

XML Schema

http://www.w3.org/TR/NOTE-ddml

The current slides reflect the W3C specification

#### Selected References

- Microsoft developed an early specification for IE5.
   Many books reference these specifications and some systems such as IE5 will probably continue to use them for a while
- This presentation looks to the W3C specification:
  The requirements for XML Schema were set out in:
  - http://www.w3.org/TR/NOTE-xml-schema-req
     A primer on XML Schema may be found at:
  - http://www.w3.org/TR/xmlschema-0/\_\_\_\_\_
  - The structural schema specification may be found at:
  - The datatype specification may be found at:
  - The RDF specification may be found at:

September 4, 2001

September 4, 2001

#### **Review of XML Parsing**

XML Schema

- What is happening when an XML parser is invoked can be confusing.
- Parsers can be validating or non-validating. All parsers check for to make sure XML documents are well formed. This does not require a DTD or schema. The document instance is simply checked to make sure it follows the syntax rules
- Validation of a document requires that the document instance have an associated DTD or schema against which it can be checked.

XML Schema

## Schema Benefits

- Schema are more powerful than DTDs in that they:
  - Allow for inheritance (namespaces)
  - Allow modular construction
  - Provide a mechanism to avoid name collisions (namespaces)

XML Schema

- Allow content control (datatypes)
- Allow for more documentation in content description (schema)
- Allow for simpler parsing (schema)

#### September 4, 2001

3



#### Namespace Association(1)

- A simple association between an element and a namespace would appear as shown below: <html xmlns="http://www.w3.org/TR/WD-HTML40">
- This says that the element "html", and the allowable sublements are defined by the named attribute. (In this case, "WD-HTML40" is actually a specification and not a schema)
- The namespaces are extensible and can be combined a number of ways. As used above, the namespace applies to all children or subelements of the element "html"
- If the namespace association is "unqualified", as the example above is, subelements of the element would appear as in previous versions of XML

XML Schema

September 4, 2001

#### Namespace Association(2)

- Formally "xmlns" is the DefaultAttName.
- The namespace specification also allows for a PrefixedAttName which is "xmlns" followed by a ":" followed by an "NCName".
- An NCName begins with a letter or underscore and that has a few other restrictions in terms of allowed symbols in the name.
- The NCName is used as a prefix for elements from that namespace – including the one for which it is an attribute:

<mbs:email xmlns:mbs="http://www.pitt.edu/~spring/m\_schema.xsd">

September 4, 2001	XML Schema	12

## Modularity, Inheritance and **Collision Avoidance**

- Multiple namespaces may be associated with a given element and its sub elements: <html:html xmlns:html="href2" xmlns:spring="href1" xmlns:math="href3">
- allows the children of html to include elements from all
- of these namespaces <html:head><<html:title>The title of the doc</html:head>
- <hr/>
  <hr/> </html:body> </html:html>
- Note that there is no conflict between the "equation" elements from the two namespaces XML Schen

September 4, 2001

#### Scope of a Namespace

٠ Namespaces can also be scoped within a document Given a top level association, subelements may be • explicitly defined as belonging to another namespace: <email xmlns="http://www.pitt.edu/~spring/m\_schemaxsd"> <to>Joe</to>

<from>Mary</from>

<body> <eq:eqnxmlns:eq ="http://www.pitt.edu/~spring/e\_schemaxsd"> <eq:relation>some element</eq:relation></eq:eqn>

<h1>some text</h1>

</body></email>

September 4, 2001

#### Namespaces in Context

XML Schema

- Namespaces define both general schemas and XML • specific schemas such as:

  - XML Link Language (XLL) capabilities XML Style Language (XSL) capabilities XSL Transformation (XSLT) capabilities XML Pointer capabilites
- Parsers are based on specifications which imply specific schema. For example, XSLT parsers.
  - IE5 accepts XML documents that begin: <?xml version="1.0"?> <xst stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">->

  - CXSTStylesneet XminSxSE" http://www.w3.org/TRWD-xst">>>
    While James Clark's XT accepts documents that begin:
    <?xml version="1.0"?>
    <xstStylesneet xmlnsxsE" http://www.w3.org/1999/XSL/Transform">>
    http://www.w3.org/1999/XSL/Transform
    http://www.w3.org/1999/XSL/Transform
    15

A Simple Schema	
<ul> <li>A schema is an XML document where the top leve element is "schema" and the associated namesapce that for XML schema. Thus: <schema schema="" xmlns="http://www.w2.org/20010/ML"> <clement "mynote"="" name=""> <clement "mynote"="" name=""> <clement "mynote"="" name=""> <clement "to"="" name="" type="string"></clement> <element name="To" type="string"></element> <element name="Fom" type="string"></element> <element name="Note" type="string"></element> <element name="Date" type="date"></element>  <le></le></clement></clement></clement></schema></li> </ul>	
September 4, 2001 XML Schema	16

#### The Schema Referenced

Assuming the schema defined on the previous slide is located in the current directory in the file mbsnote.xsd, the following now allows validation:

<mynote xmlns = "mbsnote.xsd" xmIns:xsi ="http://www.w3.org/2001/XMLSchema-instance" xsi :schema.location="mbsnote.xsd http://www.pitl.edu/\_spring/mynote.xsd" Date = "2001-05-27">

<To>Jonathan</To> <From>Patrick</From> <Note>Here is a little message for you</Note>

September 4, 2001

</xsd:schema>

</mynote>

## A Schema using Type

XML Schema

• The "complexType" element defines structure.

- The schema namespace is qualified and the target and The schema namespace is qualified and the target a default namespace are the same <xsd:schema xminss:wd="http://www.w3.org/2001/XMLSchema." xmins= "http://localhost/" targetNamespace:" http://localhost/"> <xsd:element name = "htp://localhost/"> <xsd:element name = "mynotetype"> <xsd:sequence> <xsd:sequence> <xsd:sequence> <xsd:element name = "Tor" type = "xsd:string"/> <xsd:element name = "From" type = "xsd:string"/> <xsd:element name = "Note" type = "xsd:string"/> </xsd:sequence> <xsd:attribute name = "Date" type = "xsd:string"/> </xsd:sequence>

XML Schema

	Just for clarity	
qualified and <schema xmlms<br="">xi <cement nai<br=""><complexty <sequence <c </c </sequence </complexty </cement></schema>	element name = "To" type = "string"/> element name = "From" type = "string"/> element name = "Note" type = "string"/> ≥> name = "Date" type = "date"/>	get is
September 4, 2001	XML Schema	19

# 

XML Schema

September 4, 2001

<section-header><code-block></code>

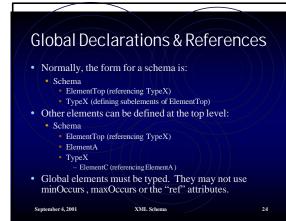
A Schema with Occurrence(3)	
<xsd:complextype name="BODYT"> <xsd:choice> <xsd:clement <br="" name="Part" type="PARTT">minOccurs="1" maxOccurs="10"/&gt; <xsd:element <br="" name="Chapter" type="CHAPTERT">minOccurs="1" maxOccurs="unbounded"/&gt; </xsd:element></xsd:clement></xsd:choice> </xsd:complextype>	
<xsd:complextype name="PARTT"> <xsd:cemplextype <br="" name="Chapter" type="CHAPTERT">minOccurs="1" maxOccurs="unbounded"/&gt;   </xsd:cemplextype></xsd:complextype>	
September 4, 2001 XML Schema	22

#### Choices, Sequences, and Sets

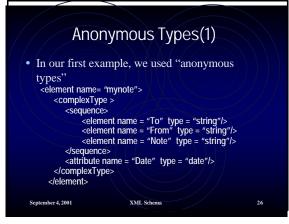
- The example above shows how minOccurs and maxOccurs can be used to control the elements in an instance
- Sequence and choice elements can also be used and nested in a variety of ways.
- Schema also allows, with some restrictions, and "all" element that says all the elements named must appear, but they may appear in any order. The restrictions are:
  - It must occur at the top level of the schema
  - The occurrence indicators can only be "0" or "1"The all group may not be nested in a sequence or choice

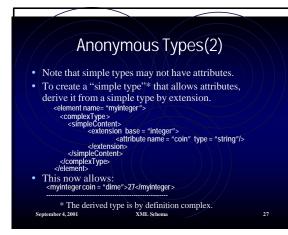
XML Schema

September 4, 2001



Example of a Global Element an	d
Reference	
<xsd:schema xmlns:xsd="btip://www.w3.org/2001/XMLSchema.&lt;br">xmlns="http://localhost/" units="http://localhost/"</xsd:schema>	
<pre>targetNamespace=" http://docalbost/"&gt;</pre>	
<xsd:element name="comment" type="xsd:string"></xsd:element> <xsd:complextype name="mydoctype"> <xsd:sequence></xsd:sequence></xsd:complextype>	
<pre><xsd:element minoccurs="0" ref="comment"></xsd:element> <xsd:element name="FrontMatter" type="FMT"></xsd:element></pre>	
<pre><xsd:element name="Body" type="BODYT"></xsd:element> <xsd:element name="EndMatter" type="EMT"></xsd:element></pre>	
September 4, 2001 XML Schema	25





#### Mixed Content Models

- To this point, all elements have been made up of:
   elements only
   data only
- An element that contains both is said to have a mixed content model.
- Schema allow mixed content via a complex type.
- Unlike XML 1.0 which allowed subelements and data to be randomly intermingled, the schema specification says the order of the subelements must be as specified by the complex type. An example is shown on the next slide

XML Schema

28

29

September 4, 2001

# 

<xsd:sequence> <xsd:element name = "emph" type = "xsd:string"/> <xsd:element name = "warning" type = "xsd:string"/> </xsd:sequence> </xsd:celement> <xsd:attribute name = "Num" type = "xsd:integer"/> //xsd:sequence>

XML Schema

- </r>
  Asd:complexity
  /xsd:element>
  September 4, 2001

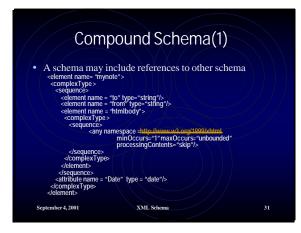
# Mixed Content Example

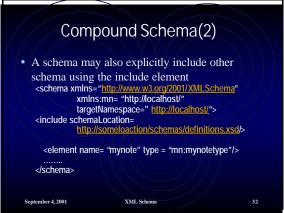
- Given the declaration on the previous slide, the following instance is now valid: <example Num = "23">
  - <Title>Here is another example</Title>
  - <Explan>
  - Here is some <emph> mixed content</emph> and a <warning>caution about being careful when using and declaring these.</warning> </Explan>

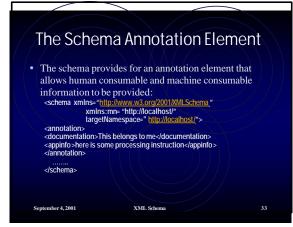
XML Schema

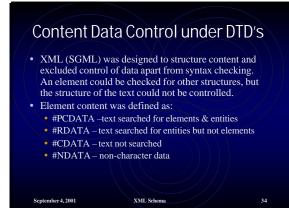
. </example>

September 4, 2001









#### Attribute Value Control under DTDS • Attribute values could be somewhat more finely controlled, most notably as enumerated values, but the control was still generally limited to very broad classes • Some of the built in datatypes for XML for

- Some of the built in datatypes for XML for attributes included
  - ID
  - IDREF
  - NMTOKEN
  - NOTATION

September 4, 2001

#### DataTypes under Schema

XML Schema

- If XML was to be used for e-commerce, and as a wrapper for data interchange, more control was needed
- XML schema are designed to allow detailed control of element content and attribute values.
- All of the DTD controls (PCDATA, RDATA, and CDATA are replaced with the string type
- Schemas have a total of 44 built in primitive(19) and derived(25) data types versus the 10 under DTD's (most of which were for attribute values
- Schemas also provide powerful tools for defining additional user defined derived data types

XML Schema

36

35

#### PrimitiveDataTypes

- String "Hello"
- Boolean {true, false}
- Decimal 9.4 •
- Float 13.78E1
- double 13.78E1
- duration P2Y2M4Detc
- dateTime CCYY-MM-
- DDhh-mm-ss
- Time hh:mm:ss.sss
- Date CCYY-MM-DD

September 4, 2001

- gYearMonth CCYY-MM gYear CCYY
  - gMonthDay MM-DD
  - gDay DD
  - gMonth MM
  - hexBinary 00A8
  - base64Binary dPm6
  - anyURI http://xyz.com
  - Qname a qualified namespace name
  - NOTATION a NOTATION from XML

#### **Derived DataTypes**

XML Schema

- normalizedString A string with tabs, line feeds, Token A normalized string with consecutive spaces reduced to one
- Language one of the established language codes
- IDREFS Attributes only, same as previously defined
- NMTOKEN Attributes only, same as previously defined NMTOKENS Attributes only, same as previously defined
- NCName a namespace name
- ID Attributes only, same as previously defined IDREF Attributes only, same as previously defined

September 4, 2001

#### **Derived DataTypes**

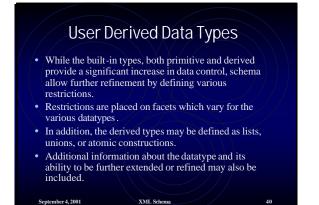
XML Schema

- Integer decimal with no fractional part nonPositiveInteger 0 to negative infinity negativeInteger 1 to negative infinit Long the integers from-2223372036854775808 to 9223372036854775807 int the integers from-2147483647 to 2147483647

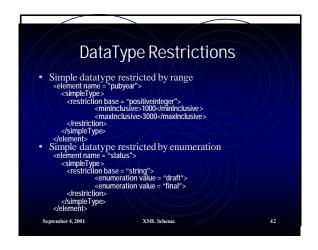
- Short the integers from -32768 to 32767 Byte the integers from -128 to 127 nonNegativeInteger integers greater than or equal to 0
- nonvegativenteger integers greater than of equal to 0
   unsignedLong non negative integer less than 18446744073709551615
   unsignedInt unsignedLong less than 4294967295
   unsignedShort unsignedInt less than 65535
   unsignedByte unsignedShort less than 255
   positiveInteger nonNegativeInteger gretaer than or equal to 1

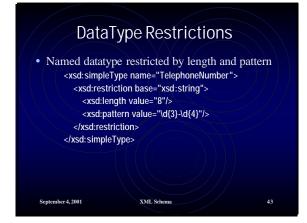
XML Schema

#### September 4, 2001



**Constraining Facets**  Examples of Constraining facets for a couple primitive types would include the following: Typè Facets length, minLength, maxLength, pattern, string enumeration, whiteSpace boolean pattern, whiteSpace totalDigits, fractionDigits, pattern, whiteSpace, decimal enumeration, maxInclusive, maxExclusive, minInclusive, minExclusive September 4, 2001 XML Schema





#### The Schema in Overview

- The schema element serves as a logical container for:
   type definitions: A set of named simple and complex type definitions.
  - attribute declarations: A set of named (top-level) attribute declarations.
  - element declarations: A set of named (top-level) element declarations.
  - attribute group definitions: A set of named attribute group definitions.
  - model group definitions: A set of named model group definitions.
  - notation declarations: A set of notation declarations.

annotations: A set of annotations.
September 4, 2001 XML Schema