Introduction to XML

Prof. Vipul Dabhi
Department of Information Technology,
D. D. University.

What Is Markup?

- Information added to a text to make its structure comprehensible
- Pre-computer markup (punctuational and presentational)
  - Word divisions
  - Punctuation
  - Copy-editor and typesetters marks
  - Formatting conventions

Computer markup

- Any kind of codes added to a document
  - Typesetting (presentational markup)
    - MS Word and its ilk, TeX, Scribe, Lout, Script, nroff, XYVision
  - Declarative markup
    - HTML (sometimes)
    - XML

What is XML?

- a meta language that allows you to create and format your own document markups
- a method for putting structured data into a text file; these files are
    - easy to read
    - unambiguous
    - extensible
    - platform-independent

Comparisons: HTML Vs XML

XML

- Extensible set of tags: allows user to specify what each tag and attribute means
- Content orientated
- Standard Data infrastructure
- Allows multiple output forms
- content and format are separate; formatting is contained in a stylesheet

HTML

- Fixed set of tags: tags and attributes are pre-determined and rigid
- Presentation oriented
- No data validation capabilities
- Single presentation
- content and formatting can be placed together

  <p><font=”Arial”>text

Weaknesses of HTML

- Fixed set of tags
- Predefined semantics for each tag
- Predefined data structure
- No formal validation
- Does not support semantic search
- Based solely on appearance and not on content
- Won’t do complex document

HTML Problems

- Desire for ***personalized*** tags
- Want to put data into HTML form
  - mathematics, database entries, literary text, poems, purchase orders ….
- HTML just isn't designed for that!

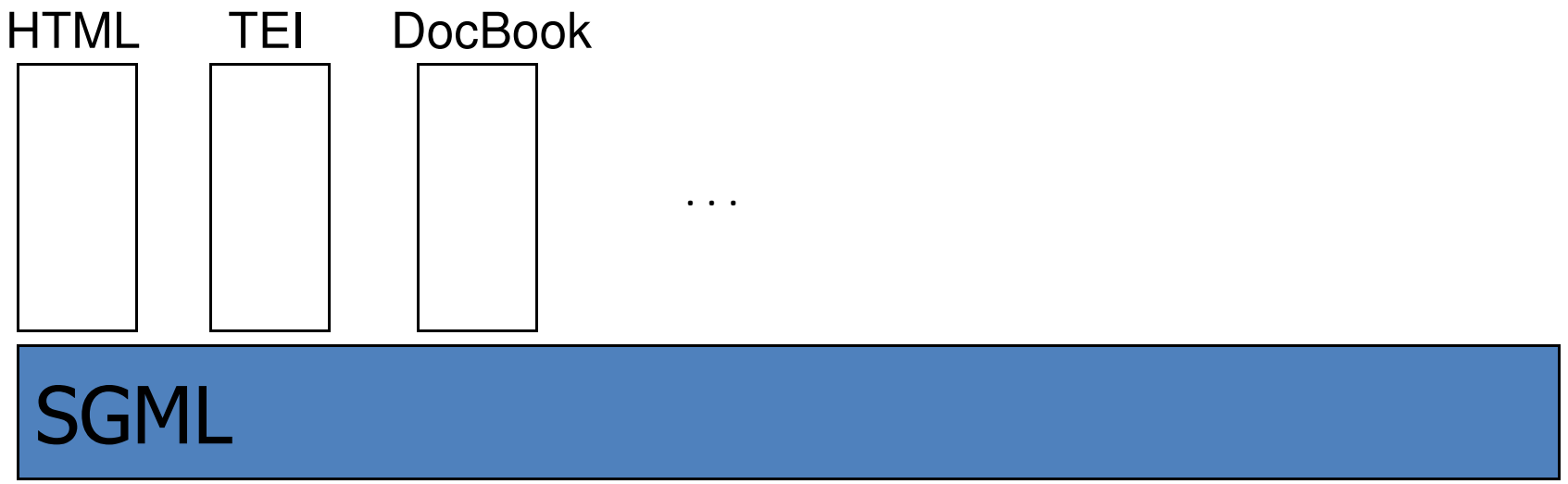
Idea: Back to the Basics

- HTML was defined using ***SGML***
  - ***S***tandard ***G***eneralized ***M***arkup ***L***anguage
  - A meta-language for defining languages.

- Complex, sophisticated, powerful

- Idea: Use ***SGML***

Languages based on SGML



Idea : "Webified" SGML

- New e**X**tensible **M**arkup **L**anguage: ***XML***
- Can use ***XML*** to define new languages
- ***Distributes*** easily on the Web
- Can ***mix*** different types of data together

XML Design Goals

- XML can be used straightforwardly over internet
- It shall be easy to write programs which process XML document
- XML design should be prepared quickly
- XML shall be compatible with SGML
- XML shall support a wide variety of applications
- XML documents should be human-legible and reasonably clear.

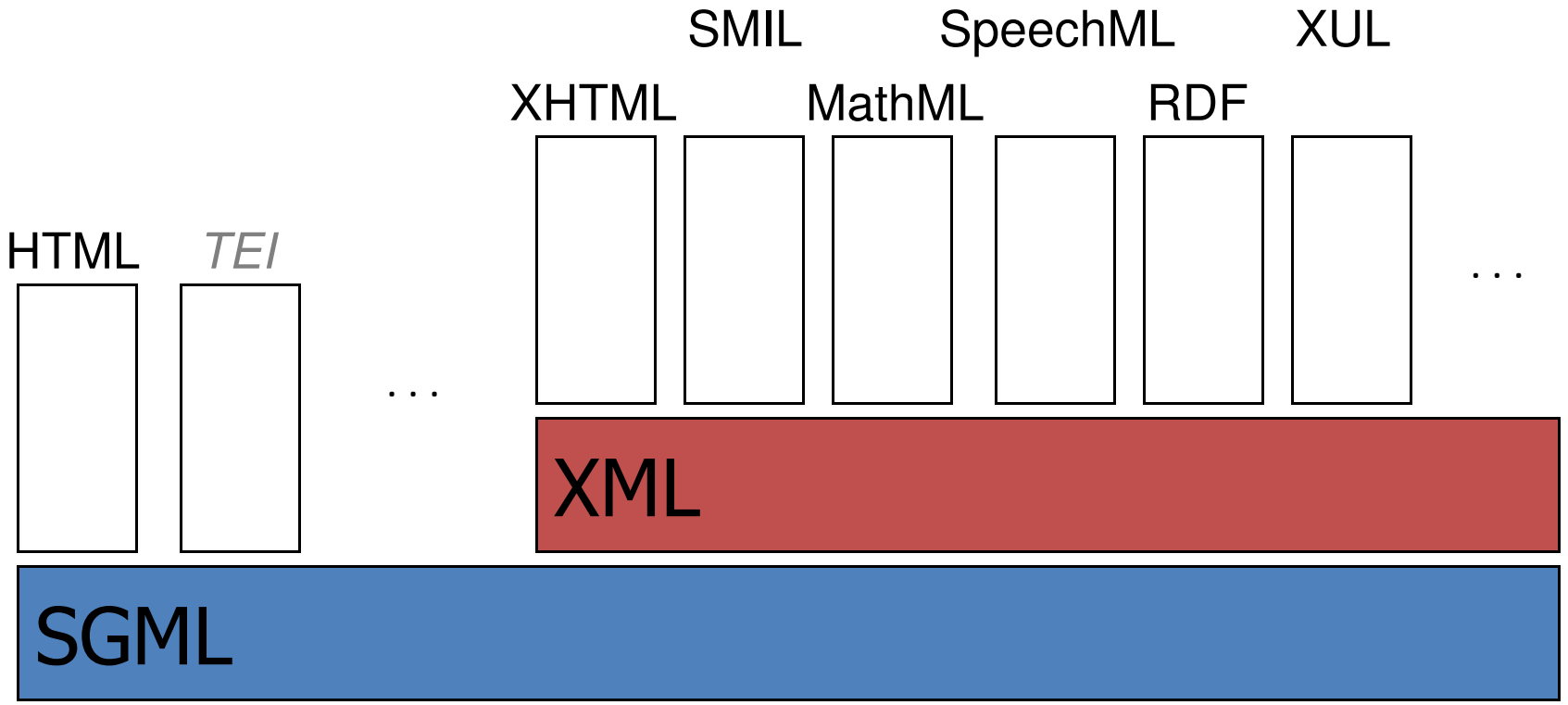
Key features of XML

- Extensibility
- Media and Presentation independence
  - Separation of contents from presentation
- Structured
- Validation

Evolution of XML

- Many XML languages, optimised for different roles
  - ***MathML*** -- for mathematics
  - ***SMIL*** -- for synchronised multimedia
  - ***RDF*** -- for describing “things”
  - ***XUL*** -- for describing the Navigator 5 user interface

The XML Family Tree



XML pieces

- there are 3 components for XML content:
  - the XML document
  - DTD (Document Type Declaration)
  - XSL (Extensible Stylesheet Language)
- The DTD and XSL do not need to be present in all cases

A well-formed XML document

- elements have an open and close tag, unless it is an empty element
- attribute values are quoted
- if a tag is an empty element, it has a closing / before the end of the tag
- open and close tags are nested correctly
- there are no isolated mark-up characters in the text (i.e. < > &]]>)
- if there is no DTD, all attributes are of type CDATA by default

XML Extensibility

- XML is Meta-markup language
- Define your own markup language (tags) for your own problem domain
- Tags can be more than formatting
- Tags can be anything
  - Semantic data representation
  - Business Rules
  - Data Relationship
- Many Domain specific markup language
  - Portable across various domain
    - Healthcare and insurance
    - Medicine and Chemical

XML Processing: XML Validation

- XML data is “constrained” by a Rule
  - Employee data has to have Name and Employee ID elements
  - Name has to have both FirstName and LastName
- <Employee>
  - <Name>
    - <FirstName> John </FirstName>
    - <LastName> Doe </LastName>
  - </Name>
  - <EmployeeID> 1234 </EmployeeID>
- </Employee>

XML Processing: XML Schema

- Defines Syntax
  - Structure
  - Vocabulary
- XML Document + XML schema work together
  - XML document alone has to be well formed
  - XML schema checks validity
- Earlier DTD was used to represent XML document structure

Parts of an XML Document

- The DTD / Schema
- Elements
- Attributes
- General entities
- Character references
- Comments
- Marked sections
- Processing instructions
- Notations
- Identifiers and catalogs

XML basics

- <?xml ?> the XML declaration

- not required, but typically used

- attributes include:

(a) version

(b) encoding – the character encoding used in the document . standalone – if an external DTD is required.

<?xml version=”1.0” encoding=”UTF-8”>

<?xml version=”1.0” standalone=”yes”>

XML basics

- <!-- --> comments

- contents are ignored by the processor
- cannot come before the XML declaration
- cannot appear inside an element tag
- may not include double hyphens

XML basics

- <tag> text </tag> an element

  - can contain text, other elements or a combination

  - element name:

    -must start with a letter or underscore and can have any number of letters, numbers, hyphens, periods, or underscores

  - **case-sensitive**;

  - may not start with *xml*

Authoring XML Elements

- An XML element is made up of a start tag, an end tag, and data in between.
- Example:

  <director> Matthew Dunn </director>
- Example of another element with the same value:

  <actor> Matthew Dunn </actor>
- XML tags are case-sensitive:

  <CITY> <City> <city>
- XML can abbreviate empty elements, for example:

<married> </married> can be abbreviated to

<married/>

XML basics

Elements (continued)

- can be a *parent, grandparent, grandchild, ancestor,* or *descendant*
- each element tag can be divided into 2 parts – *namespace*:*tag name*

Authoring XML Documents

- A basic XML document is an XML element that can, but might not, include nested XML elements.
- Example:

```
<books>
    <book isbn="123">
        <title> Second Chance </title>
        <author> Matthew Dunn </author>
    </book>
</books>
```

XML basics

- Namespaces:

  - not mandatory, but useful in giving uniqueness to an element

  - help avoid element collision

  - declared using the xmlns:*name*=*value* attribute; a URI is recommended for *value*

  - can be an attribute of any element; the scope is inside the element's tags

XML basics

- key=”value” an attribute
  - describes additional information about an element

<tag key=”value”> text</tag>

- value must always be quoted
- key names have same restrictions as element names
- reserved attributes are
  - xml:lang
  - xml:space

Authoring XML Elements

- An attribute is a name-value pair separated by an equal sign (=).
- Example:

  <City ZIP="94608"> Emeryville </City>
- Attributes are used to attach additional, secondary information to an element.

XML basics

- Namespaces (continued):

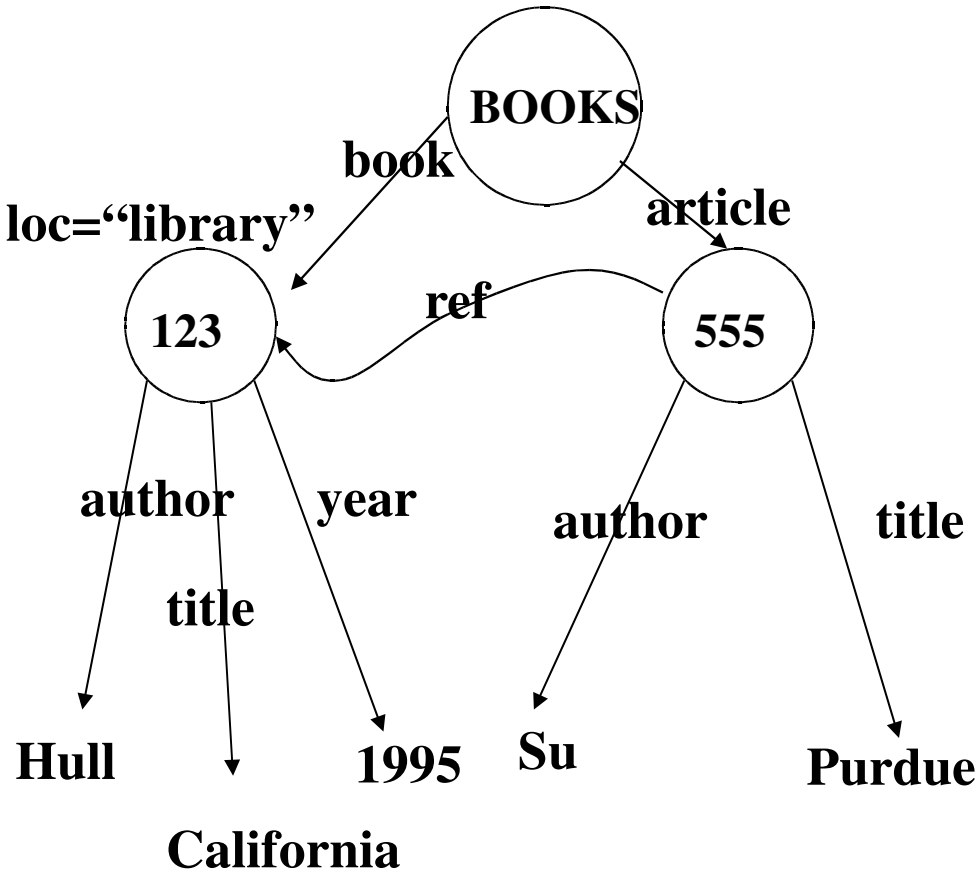
  - may define more than 1 per element

  - if no name given after xmlns prefix, uses the default namespace which is applied to all elements in the defining element without their own namespace

  - can set default namespace to an empty string to ensure no default namespace is in use within an element

XML Data Model: Example

```
<BOOKS>
<book id="123" loc="library">
  <author>Hull</author>
  <title>California</title>
  <year> 1995 </year>
</book>
<article id="555" ref="123">
  <author>Su</author>
  <title> Purdue</title>
</article>
</BOOKS>
```



Authoring XML Documents

- Authoring guidelines:
  - All elements must have an end tag.
  - All elements must be cleanly nested (overlapping elements are not allowed).
  - All attribute values must be enclosed in quotation marks.
  - Each document must have a unique first element, the root node.

Motivations of XML Schema

- Provide more powerful and flexible schema language than DTD
- Represent XML document syntax in XML language
- Support non-textual data types
  - B2B, e-Commerce
- Handle complex syntax

Document Processing

- Valid V/s Schema-Valid
  - XML schema is not part of XML 1.0
  - XML document that is validated with DTD is “Valid”
  - XML document that conforms to XML schema is “Schema-valid”
  - XML document that conforms to a particular XML schema is called “instance document” of that schema
- Definition and Declaration
  - Definition: Create new types
  - Declaration: Enable elements and attributes with specific names and types to appear in document instances.