

XML Overview

Jeff Offutt

<http://www.cs.gmu.edu/~offutt/>

SWE 432

**Design and Implementation of
Software for the Web**

Topics

1. Motivation
2. How does XML work ?
3. Syntax of XML documents
4. XML and HTML

XML

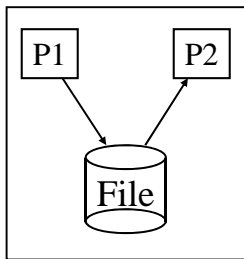
- eXtensible Markup Language
- Markup languages insert “tags” into text files to describe presentation or other information
- SGML : Standard Generalized Markup Language
 - HTML : Visual presentation
 - Latex : Document formatting
 - XML : Data description
- W3C standard: <http://www.w3.org/XML/>

Why XML?

- Passing data from one software component to another has always been difficult
- The two components must agree on format, types, and organization
- Web software applications have unique requirements for data passing:
 - Very loose coupling
 - Dynamic integration

Passing Data – 1978

- Program P2 needs to use data produced by program P1
 - Data saved to a file as records (COBOL, Fortran, ...)
 - The file format often not documented
 - Source for P1 may not be available
 - Data saved in binary mode – not readable by hand



- Format of file deduced from executing P1 and from trial and error
- MSU Computing Services, 1979 : Two weeks of trial and error executions of P1 to understand format of file

1 December 2011

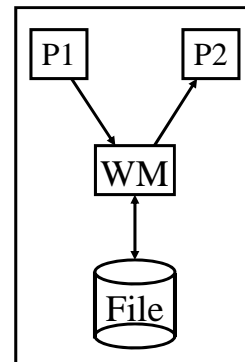
© Offutt, 2011

5

Passing Data – 1985

- Program P2 needs to use data produced by program P1
 - Data saved to a file as records (C, Ada, ...)
 - The file format often not documented
 - Data saved as plain text

- Both P1 and P2 access the file through a “wrapper module”
- Module needs to repeatedly updated
- Module written by development team
- Data hard to validate
- Mothra, 1985 : ~12 data files shared among 15 to 20 separate programs



1 December 2011

© Offutt, 2011

6

Wrapper Method Problems

- Slow – everything is a file in plain text
- Sharing – Developers of P1 and P2 must agree to share source of WM
- Maintenance – Who has control of WM?
- Solution – data sharing that is :
 - Independent of type
 - Self documenting
 - Easy to understand format
 - Especially important for web applications – XML

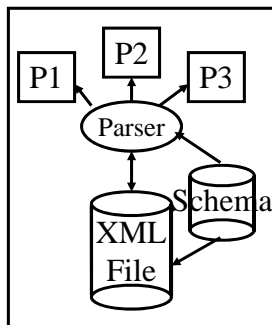
1 December 2011

© Offutt, 2011

7

Passing Data – 21st Century

- Data is passed directly between components
- XML allows for self-documenting data



- P1, P2 and P3 can see the format, contents, and structure of the data
- Free parsers are available to put XML messages into a standard format
- Information about type and format is readily available

1 December 2011

© Offutt, 2011

8

Topics

1. Motivation
2. How does XML work ?
3. Syntax of XML documents
4. XML and HTML

12/1/2011

© Ofutt

9

Introductory Example

- Programmers can create their own tags
- Tags have been designed for mathematics, formal specifications, resumes, recipes, addresses, ...
- Pizza Markup Language (PML):

```
<pizza>  
  <topping extracheese="yes">Pepperoni</topping>  
  <price> 13.00 </price>  
  <size> large </size>  
</pizza>
```

12/1/2011

© Ofutt

10

Markup Languages – Typesetting

Markup languages began in typesetting

- Documents were marked-up to represent how they would be printed
- For example, words can be **Bold**, *italicized*, or underlined
- Typesetting only effects the printing of specific phrases or words, and not categories of phrases or words

12/1/2011

© Offutt

11

Markup Languages–Semantic Tags

- Markup languages can be used to logically organize the contents of a document
- For example, a document representing a book can contain the following organizational tags :
 - Title
 - Chapter headings
 - Section headings

12/1/2011

© Offutt

12

Markup Languages–Semantic Tags

- A markup language can also provide semantic information (*meta-data*) about the text in a document
 - Examples : *First name, Last name, Phone number*
- Semantic tags can improve the accuracy of document queries
 - Documents can be searched using their tag assignments rather than the plain-text contents

12/1/2011

© Offutt

13

Markup Languages–Semantic Tags

- Use semantic tags to define the hierarchical structure of the document
 - Author
 - First name
 - Last name
 - Publisher
 - Name
 - Address

12/1/2011

© Offutt

14

Markup Languages – Examples

- Typesetting tags
 - `<bold> Chapter 1 </bold>`
 - `<italic> Background </italic>`
 - `<underline> Important text </underline>`
- Semantic tags
 - `<first name> Steffi </first name>`
 - `<last name> Offutt </last name>`
 - `<phone number> 703-123-1234 </phone number>`

12/1/2011

© Offutt

15

SGML

- SGML — Standard Generalized Markup Language
- Set up by the ISO in 1986
- Super set of all markup languages – includes all the features of every markup language derived from it
- Allows a document to be annotated with text that describes the semantic meanings of portions of the document

12/1/2011

© Offutt

16

SGML

- Separates the structure of the document from the content
 - The structure denotes the purpose of the document's data
- Use grammars (Schemas and DTDs) to define the syntax of the annotations used in a document
- SGML captures meta-data for a document by marking up the content

12/1/2011

© Offutt

17

Characteristics of XML

1. XML is extensible
 - Tags have been designed for mathematics, formal specifications, resumes, recipes, addresses, ...
2. XML has a strict structure
3. XML is validating
 - Grammars (Schemas & DTD) define XML languages
 - Documents can be checked against the grammar
 - Required fields, etc
 - Allows programs to assume the data is formatted correctly, reducing the amount of checking the program must do

12/1/2011

© Offutt

18

XML Provides Data Independence

- Allows data to be used by any application
- Requires every document to be in a clear and specific format
- Fosters information sharing better than other markup languages

12/1/2011

© Offutt

19

XML Simplifies Data Sharing

- Plain text
 - Create and edit files with any editor
 - Easy to debug
 - Scalability : suitable for both small and large scaled data
- Data identification
 - Once different parts of the information have been identified, they can be used in different ways by different applications
- Data transference
 - Very easy to move between XML and form parameters
 - Very easy to move between XML and databases

12/1/2011

© Offutt

20

XML Can Easily Be Displayed

- The stylesheet standard, XSL, lets you dictate how to format and display the data
- Since XML is inherently style free, you can use different stylesheets to produce different output formats

XML Example

```
<message>
  <to> you@yourAddress.com </to>
  <from> me@myAddress.com </from>
  <subject> XML Is Really Cool </subject>
  <text>
    How many ways is XML cool? Let me count
    the ways ...
  </text>
</message>
```

Topics

1. Motivation
2. How does XML work ?
3. Syntax of XML documents
4. XML and HTML

XML Structure

- Containment : Tags can be contained in other tags
- Tag names should be meaningful
- All tags must have an end tag
 - Note that HTML does not ... meaning HTML is not fully SGML-compliant

XML Can Easily Be Validated

- XML messages are described in grammars
- Two ways to describe an XML language
 - Schemas : Grammar plus types and facets
 - Document Type Definitions (DTD) : Older, easier to read and understand, but somewhat limited
- Documents can be checked against the grammar
- Grammar can specify that certain fields are required
- Allows programs to assume the data is formatted correctly, reducing the amount of checking the program must do

12/1/2011

© Offutt

25

Syntax of XML

- XML syntax is defined at two levels
 - General syntax : defines syntax on all XML documents
 - Correct documents said to be “well formed”
 - Specific syntax : defines syntax on a specific group of documents
 - Correct documents said to be “valid”
- Statements in an XML document
 - XML declaration – which version of XML
 - Data elements – the primary contents of the document
 - Markup declarations – instructions to XML parser
 - Processing instructions – instructions to the program

12/1/2011

© Offutt

26

XML Declaration

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
```

- `<? .. ?>` Prolog declaration
- *version*
 - Identifies the version of the XML markup language used in the data
 - This attribute is required
- *encoding*
 - Identifies the character set used to encode the data
 - "ISO-8859-1" is "Latin-1" the Western European and English language character set
 - Default is compressed Unicode: UTF-8
- *standalone*
 - Tells whether or not this document references an external entity or an external data type specification
 - If there are no external references, use "yes"

12/1/2011

© Offutt

27

XML Data Element Names

- Must start with a letter or underscore, and can include digits, hyphens, and periods
- XML names are case sensitive
 - `lastName`, `lastname`, `LASTNAME` are all different

12/1/2011

© Offutt

28

XML General Syntax Rules Well-formed

- Every XML document has a single root element
 - Opening tag must be first line of XML
 - All other elements are nested inside the root element
- XML tags are surrounded by pointy brackets “< >”
- Every XML tag must have a closing tag
 - If no content: <empty/>
- XML elements must be properly nested
 - <l> ... </l> is **not well formed** XML
- All attribute values must be enclosed in quotes

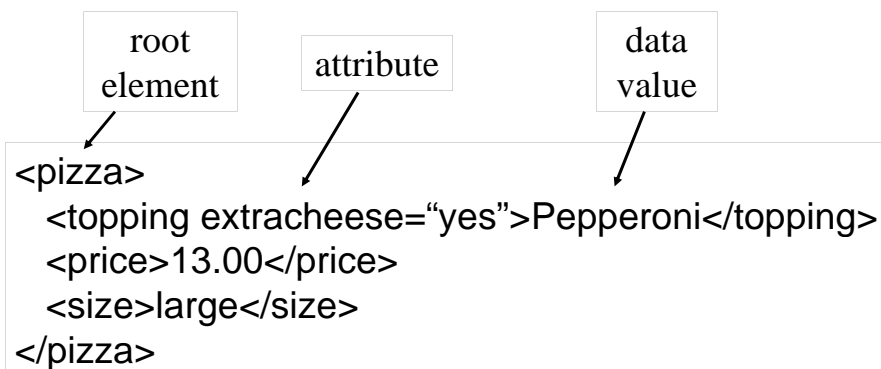
12/1/2011

© Offutt

29

XML Example

Pizza Markup Language (PML)



12/1/2011

© Offutt

30

XML Attributes

Product XML

```
<products>
  <product>
    <name>Monitor</name>
    <!-- Price can be USD, Euro, or Yuan -->
    <price currency="USD">200</price>
  </product>
</products>
```

12/1/2011

© Offutt

31

Attributes vs. Nested Tags

- In PML, “extraCheese” could have been defined as attribute or a nested tag
- Images can only be attributes
- It is easier to add new tags than attributes
- Attributes cannot define structure

```
<... name="Yao Ming">
<name>Yao Ming</name>
<name>
  <familyName>Ming</familyName>
  <givenName>Yao</givenName>
</name>
```

12/1/2011

© Offutt

32

Attributes vs. Nested Tags (2)

- Attributes are necessary when :
 - Identifying numbers or names of elements
 - Values are selected from a finite set
- Attributes should be used when :
 - No substructure
 - Attribute describes information about the element

12/1/2011

© Offutt

33

XML Entity References (Variables)

- Entities are usually used to embed special characters into XML messages
- *Document Entity* : The file that represents the document
- Other entities have names
- Entity names start with letters, dash, colon
 - Can also contain digits, periods, underscores
- References to entities surround name with &;
 - &entityName;
- Some built-in XML entities: < > & " '
- Use entities to avoid malformed XML
 - <pred> X < Y </pred> ... <pred> X < Y </pred>

12/1/2011

© Offutt

34

Topics

1. Motivation
2. How does XML work ?
3. Syntax of XML documents
4. XML and HTML

12/1/2011

© Offutt

35

XML vs. HTML

- Unlike HTML, XML tags tell you what the data means, rather than how to display it
- XML elements must be strictly nested, XML can represent data in any level of complexity
- Both XML and HTML allow empty tags; in XML an empty tag must be followed by a forward slash: `<emptyTag/>`

12/1/2011

© Offutt

36

XML vs. HTML

- XML attribute values must be surrounded by single or double quotes
- HTML does not require quotes for single values
- XML tags are case sensitive
- HTML tags are not
 - This is really confusing at first !

12/1/2011

© Offutt

37

XML Summary

- XML gives software engineers an incredibly flexible, simple, and powerful way to represent data
 - Works with all sorts of data
 - Maps naturally to tables, spreadsheets and databases
- Grammatical rules can be defined (next slides ...)
- Human readable
- Performance costs
 - Plain text files use more space on disk
 - Takes time to read, write, and reformat XML to and from internal representations
 - This cost is seldom important and almost never within web applications

12/1/2011

© Offutt

38