Institute of Public Administration

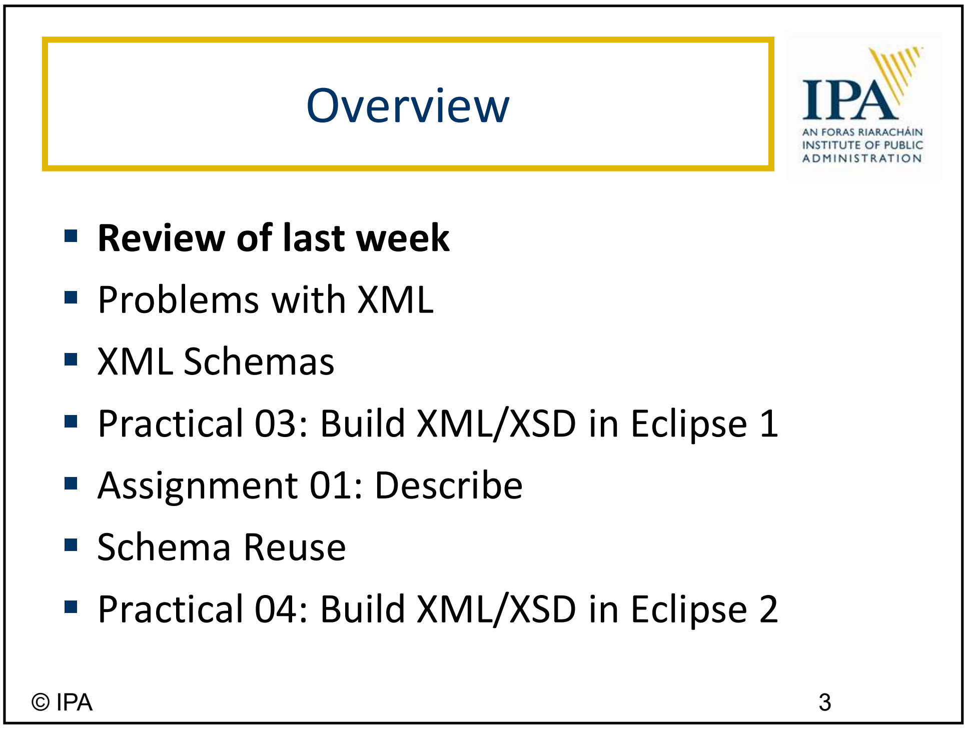
COMP41530 - Web Services in Cloud Computing

Barry Corish
Associate Lecturer, IPA

Lecture 04

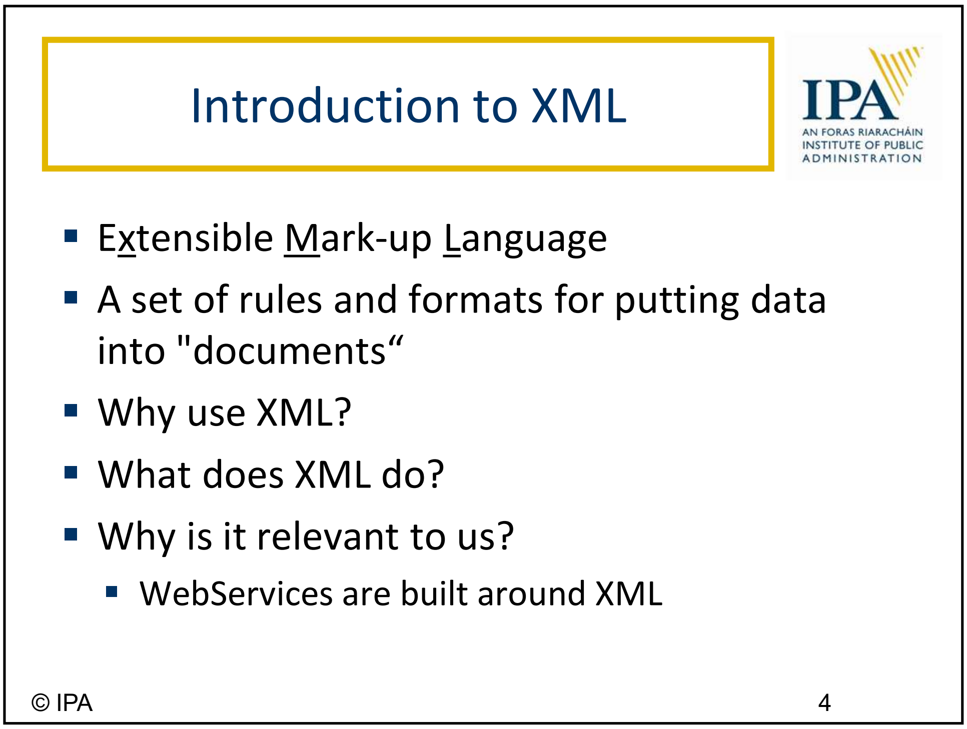
Overview

- Review of last week
- Problems with XML
- XML Schemas
- Practical 03: Build XML/XSD in Eclipse 1
- Assignment 01: Describe
- Schema Reuse
- Practical 04: Build XML/XSD in Eclipse 2



Overview

- **Review of last week**
- Problems with XML
- XML Schemas
- Practical 03: Build XML/XSD in Eclipse 1
- Assignment 01: Describe
- Schema Reuse
- Practical 04: Build XML/XSD in Eclipse 2



Introduction to XML

- Extensible Mark-up Language
- A set of rules and formats for putting data into "documents“
- Why use XML?
- What does XML do?
- Why is it relevant to us?
  - WebServices are built around XML

Some sample XML…

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Jane</to>
  <from>Dave</from>
  <subject>Reminder</subject>
  <body>Remember the milk.</body>
</note>
```

“Parts” of XML…

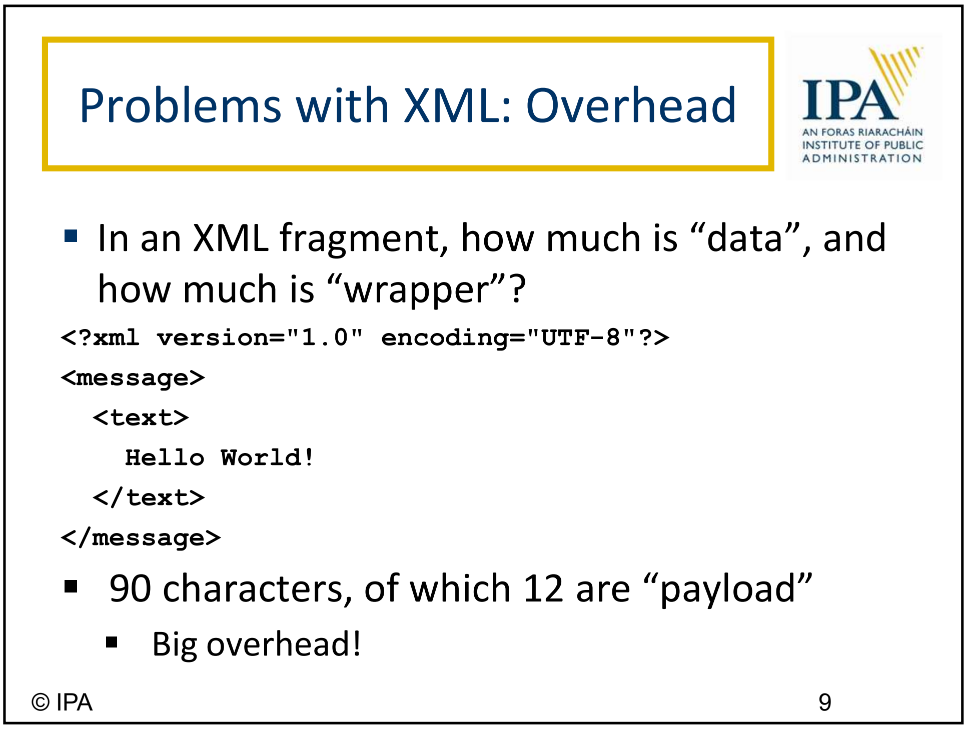
- Elements
  - Root Elements
- Attributes
  - Attributes in element start tags
- Nesting of elements
  - “Parent” and “child”
- Namespaces
  - Between documents
  - Within one document

Other things worth remembering…

- XML is about describing the data
  - Vs. HTML - about visual layout
- XML parsers are strict:
  – No mercy!
- Should use descriptive names
  - Should be human readable
- Encoding special characters

Overview

- Review of last week
- **Problems with XML**
- XML Schemas
- Practical 03: Build XML/XSD in Eclipse 1
- Assignment 01: Describe
- Schema Reuse
- Practical 04: Build XML/XSD in Eclipse 2

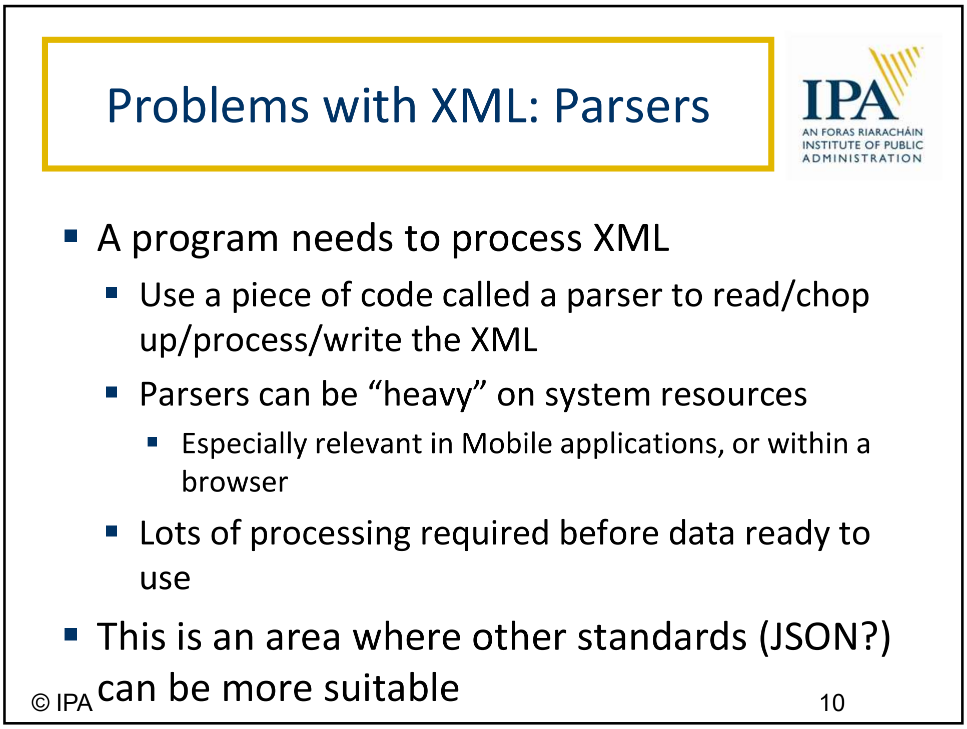


Problems with XML: Overhead

- In an XML fragment, how much is "data", and how much is "wrapper"?

```
<?xml version="1.0" encoding="UTF-8"?>
<message>
  <text>
    Hello World!
  </text>
</message>
```

- 90 characters, of which 12 are "payload"
  - Big overhead!



Problems with XML: Parsers

- A program needs to process XML
  - Use a piece of code called a parser to read/chop up/process/write the XML
  - Parsers can be "heavy" on system resources
    - Especially relevant in Mobile applications, or within a browser
  - Lots of processing required before data ready to use
- This is an area where other standards (JSON?) can be more suitable

Problems with XML: Flexibility

- On it’s own, XML is entirely flexible
  - But to process incoming XML from a variety of sources, we need it to conform to some standard.
  - Can’t work if everyone uses different attribute names
- There is a solution to this one
  - …and that’s what we’re covering next

Overview

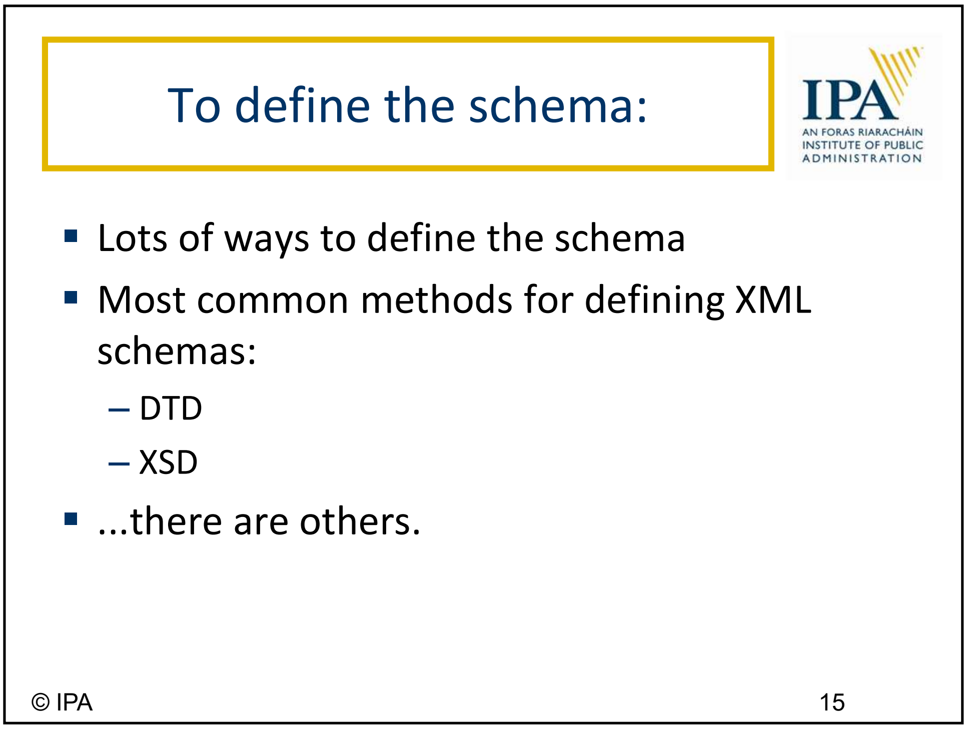
- Review of last week
- Problems with XML
- **XML Schemas**
- Practical 03: Build XML/XSD in Eclipse 1
- Assignment 01: Describe
- Schema Reuse
- Practical 04: Build XML/XSD in Eclipse 2

Defining rules for XML Documents

- XML is flexible
- We need to be able to define/limit what is acceptable in a given XML Structure
- These limits are called the "schema"
- “Validate” an XML document against the relevant schema
  - If the XML meets the rules defined in the schema, it is “validated”.

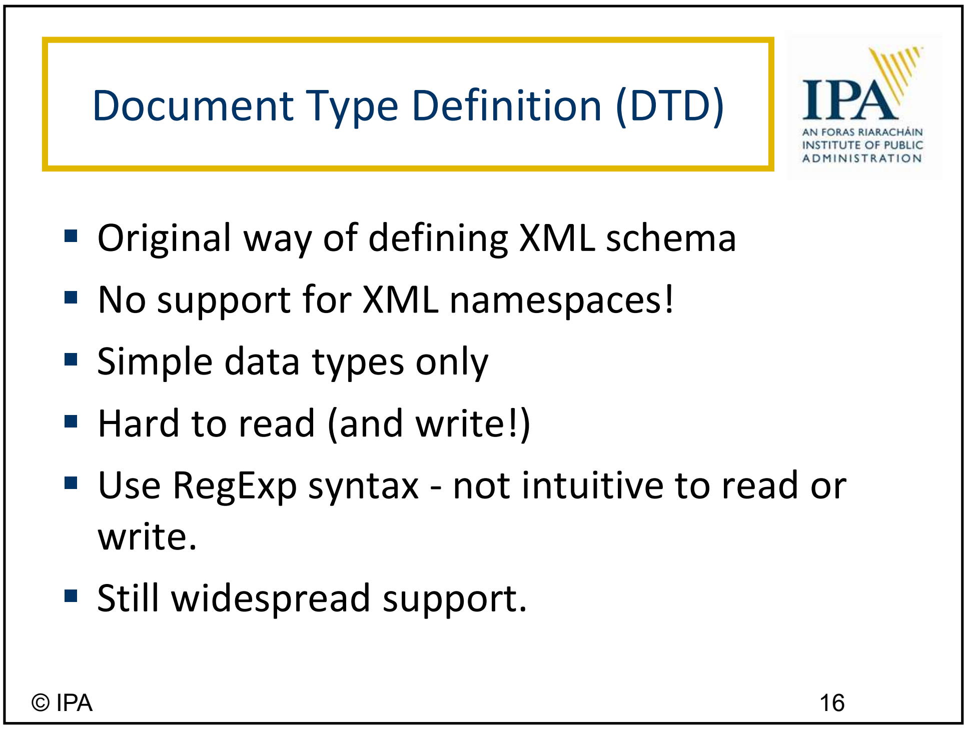
XML Schemas

- Agreed standards for format and syntax of data
- Hundreds defined and freely available across many industries.
- These are XML "Languages" e.g.:
  - RSS
  - ATOM
  - SOAP
  - XHTML
- Also being used in "documents":
  - Microsoft Office Open XML standard
  - OpenOffice



To define the schema:

- Lots of ways to define the schema
- Most common methods for defining XML schemas:
  - DTD
  - XSD
- ...there are others.



Document Type Definition (DTD)

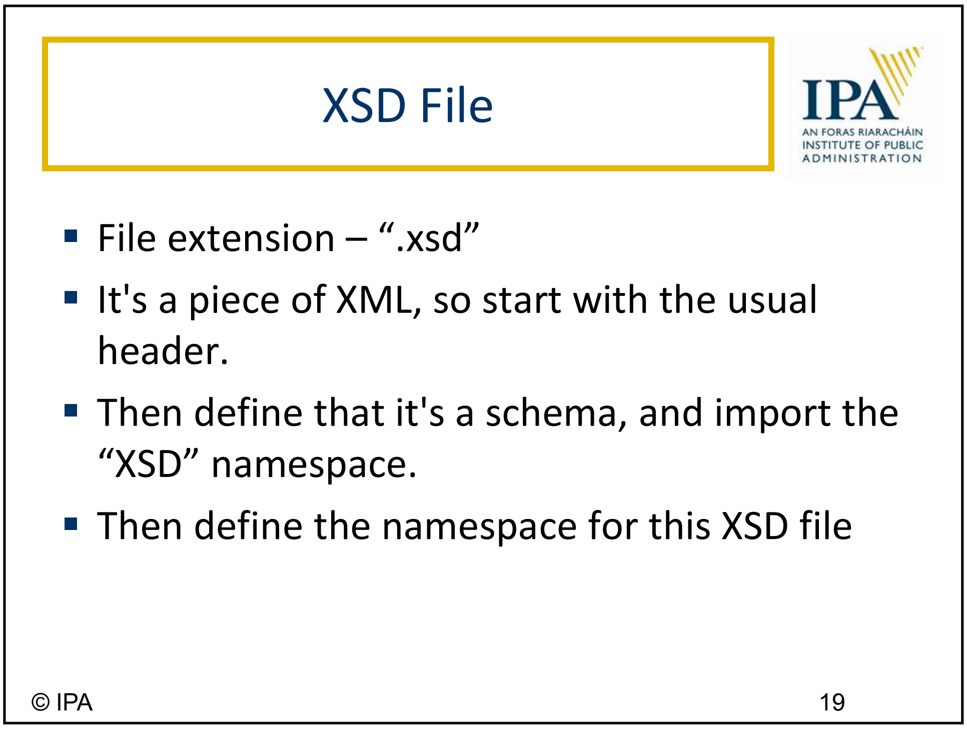
- Original way of defining XML schema
- No support for XML namespaces!
- Simple data types only
- Hard to read (and write!)
- Use RegExp syntax - not intuitive to read or write.
- Still widespread support.

XML Schema Definition Language (XSD)

- W3C standard
- Successor to DTD
- Better data types
- More flexible
- XSDs are XML based!
- Rarely written up by hand
  - Normally use tools to produce
- Almost universal support now

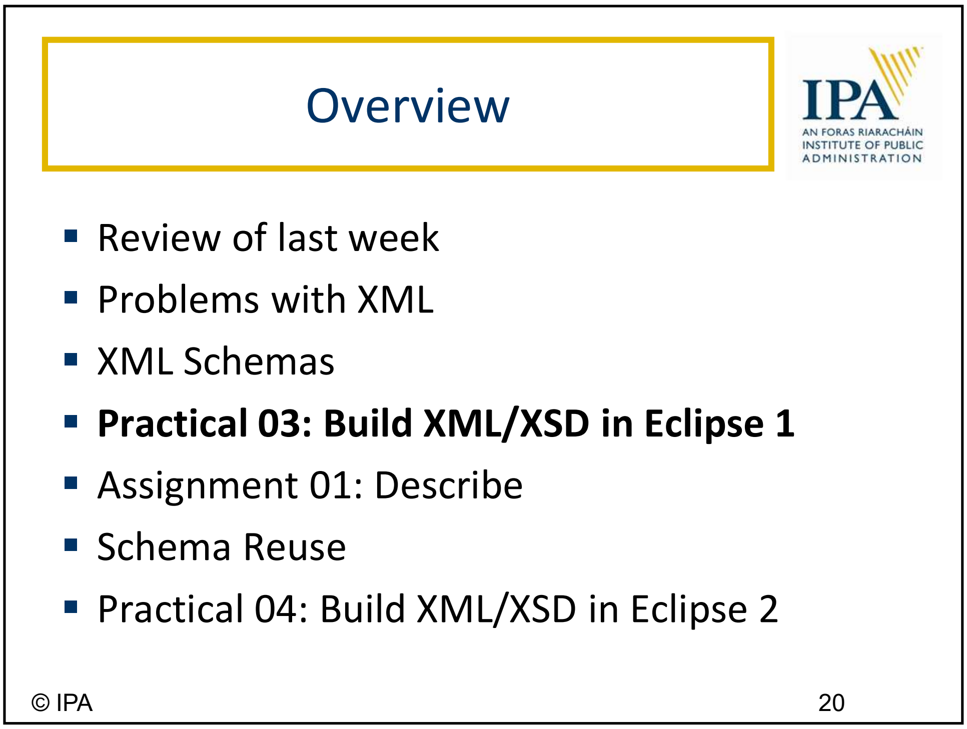
XSD Schema can restrict:

- Naming and structure (child/parent relationship) of elements and attributes
- Data types
- Maximum and or minimum number of times an element or attribute appears (aka “cardinality)
- Maximum and or minimum values for an attribute
- Force selection of the value of an attribute from a defined list
- …and much more



XSD File

- File extension – “.xsd”
- It's a piece of XML, so start with the usual header.
- Then define that it's a schema, and import the “XSD” namespace.
- Then define the namespace for this XSD file



Overview

- Review of last week
- Problems with XML
- XML Schemas
- **Practical 03: Build XML/XSD in Eclipse 1**
- Assignment 01: Describe
- Schema Reuse
- Practical 04: Build XML/XSD in Eclipse 2

Build XSD and XML for an IPA course

- Define a schema that defines an IPA course.
  - Course name
  - Date of first lecture
  - One lecturer, one or more students
  - Etc.
- Produce a piece of XML, filled with “example” data that validates against that schema.
- Two matching files, one .xsd file, one .xml file

Overview

- Review of last week
- Problems with XML
- XML Schemas
- Practical 03: Build XML/XSD in Eclipse 1
- **Assignment 01: Describe**
- Schema Reuse
- Practical 04: Build XML/XSD in Eclipse 2

Assignment 1

- Question is up on Moodle now!
- Must write matching XSD and XML files.
- XML must validate against XSD
- Submit two files by email:
  - To barry.corish@gmail.com
  - Deadline as per the question document
  - Attach both files (.xml and .xsd)

Overview

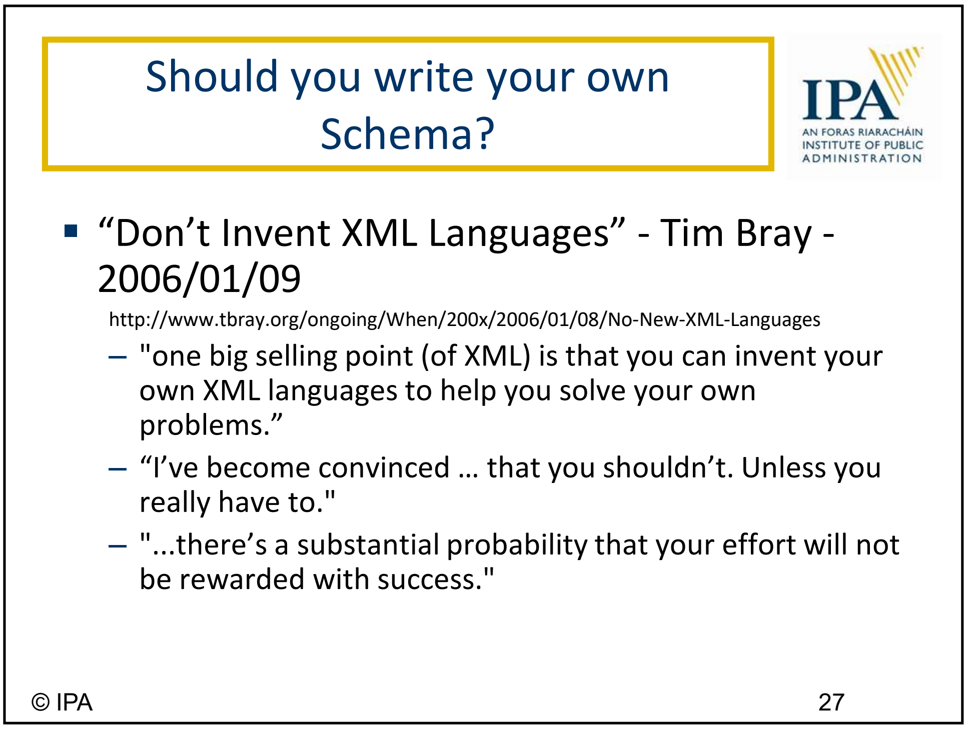
- Review of last week
- Problems with XML
- XML Schemas
- Practical 03: Build XML/XSD in Eclipse 1
- Assignment 01: Describe
- **Schema Reuse**
- Practical 04: Build XML/XSD in Eclipse 2

Available Schemas

- There's a Schema for just about everything...
  - Unless you're inventing a new business process
  - ...someone has done it before
  - ...and has written a schema for it
- Trade groups, standards bodies etc.
- Particularly important where dealing with outside organisations

Schema Re-use

- You can bring in other schemas
  - Use them as is
  - Modify
  - Extend
  - Restrict

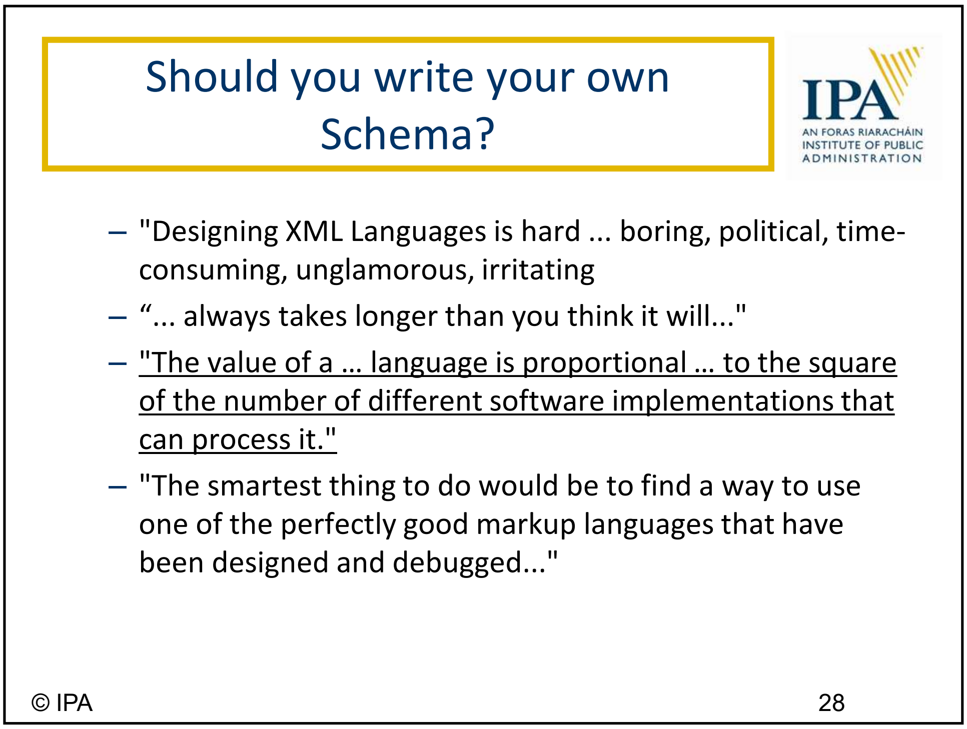


Should you write your own Schema?

- “Don’t Invent XML Languages” - Tim Bray - 2006/01/09

  http://www.tbray.org/ongoing/When/200x/2006/01/08/No-New-XML-Languages

  - "one big selling point (of XML) is that you can invent your own XML languages to help you solve your own problems.”
  - “I’ve become convinced … that you shouldn’t. Unless you really have to."
  - "...there’s a substantial probability that your effort will not be rewarded with success."



Should you write your own Schema?

- "Designing XML Languages is hard ... boring, political, time-consuming, unglamorous, irritating
- “... always takes longer than you think it will..."
- "The value of a … language is proportional … to the square of the number of different software implementations that can process it."
- "The smartest thing to do would be to find a way to use one of the perfectly good markup languages that have been designed and debugged..."

Should you write your own Schema?

- Things have changed a little since 2006
  - If you need to, you need to
- Tim Bray’s points still stand:
  - This is for integration
    - Make sure the people you want to integrate with are happy with this Schema
  - But do some research first
    - See has someone else done the hard work for you

Overview

- Review of last week
- Problems with XML
- XML Schemas
- Practical 03: Build XML/XSD in Eclipse 1
- Assignment 01: Describe
- Schema Reuse
- **Practical 04: Build XML/XSD in Eclipse 2**

Practical 04

- “Music Collection”
  - We have a collection of music in various formats: CD, Vinyl, Tape, Digital
  - Single tracks and albums
  - We have playlists
  - Define an XML structure to hold this information
  - Produce and XSD to enforce the structure

Overview

- Review of last week
- Problems with XML
- XML Schemas
- Practical 03: Build XML/XSD in Eclipse 1
- Assignment 01: Describe
- Schema Reuse
- Practical 04: Build XML/XSD in Eclipse 2