A Manager's Introduction to XML

Mulberry Technologies
17 West Jefferson St.
Suite 207
Rockville MD 20850
Phone: 301/315-9631
Fax: 301/315-8285
info@mulberrytech.com
`http://www.mulberrytech.com`

Version 1.0

Mulberry
Technologies, Inc.

A Manager's Introduction to XML

A little background

A little history

Significance of standard syntax

What can you do with data like this?

For more information

Welcome to XML 2000

Administrivia

- Start, end, break
- How this will work
- Questions are always welcome
- Anything else?

Where we are going

- Look at XML as a standard for *markup languages*
    - With some attention to history of these technologies
    - ...And some attention to what it does, what it doesn't do, and why this is important
- Distinguish between
    - XML (XML 1.0)
    - XML-related standards and technologies
    - XML applications
- Consider XML as a foundation for networked information services
    - Different applications, media types
    - Different kinds of data
    - On unified, non-proprietary platform

Where we are *not* going

- No detailed look at XML syntax
- No in-depth consideration of any one application or market
    - EDI, publishing, syndication, portals....
- No explicit coverage of XML tools, APIs, software packages

A quick poll

- How many are interested in using XML for
    - Publishing?
    - Business-to-business?
    - Business-to-customer?
    - Internal systems/processes?
- Who would describe themselves as
    - Manager or planner or strategist?
    - Analyst or developer?
    - Writer, researcher, consultant?
- How many
    - Know HTML (hands-on)?
    - Know SGML or XML?

What is XML?

XML is two things:

- Standard syntax for markup languages
- A data modeling language for markup languages in XML syntax

What is XML *not*?

- Anything not defined in the XML 1.0 Recommendation
  - Specified by W3C (World Wide Web Consortium) February 1998
- This includes all the XML “related technologies”
  - XSL (Extensible Stylesheet Language)
  - XLink, XPointer
  - XML-Data, RDF (Resource Description Framework)
  - etc. etc.

What is a markup language?

- Evolves out of publishing, documentation applications
- Solution to problem of managing structured & semi-structured data electronically
- (May be) *vendor- and application-independent*

A little background

What is markup?

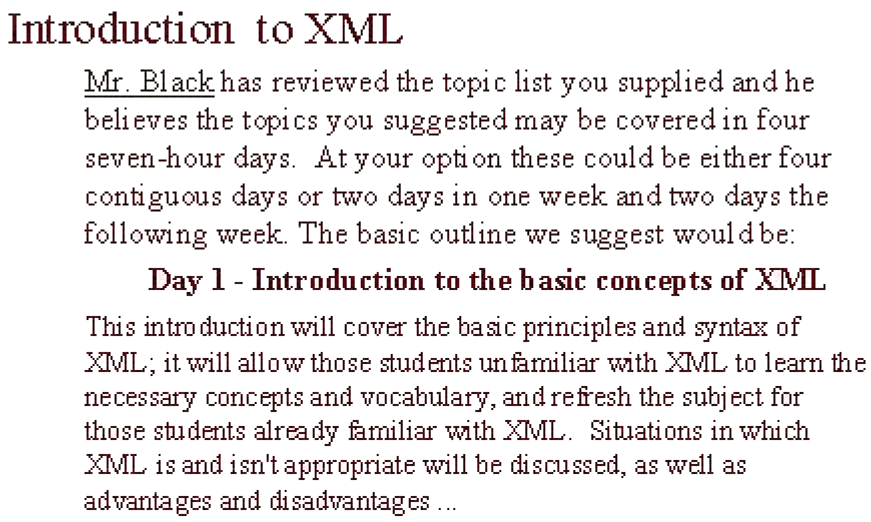
Data added to data to make it more useful:

- “Text is never *just* text”
  `Text is never <stress>just</stress> text`

Format-based markup

- Typically used in:
    - Word-processors
    - Desktop publishing systems
    - Composition systems
    - Display engines (such as browsers)
- In practice, HTML is format-based

Sample formatted document



Introduction to XML

Mr. Black has reviewed the topic list you supplied and he believes the topics you suggested may be covered in four seven-hour days. At your option these could be either four contiguous days or two days in one week and two days the following week. The basic outline we suggest would be:

Day 1 - Introduction to the basic concepts of XML

This introduction will cover the basic principles and syntax of XML; it will allow those students unfamiliar with XML to learn the necessary concepts and vocabulary, and refresh the subject for those students already familiar with XML. Situations in which XML is and isn't appropriate will be discussed, as well as advantages and disadvantages ...

The sample In RTF

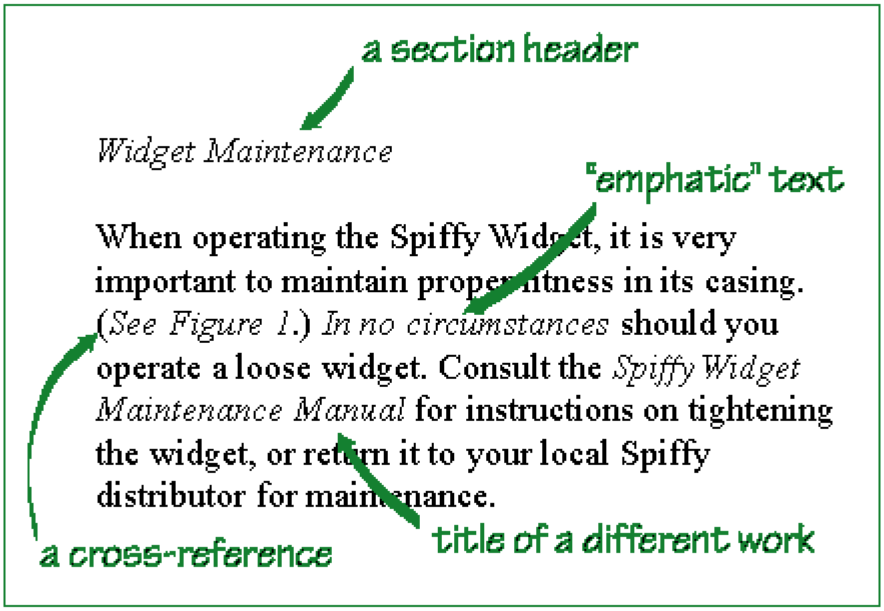
RTF is used by Microsoft Word (and in other word processors)

```
\pard\plain \sa240\widctlpar \f4\fs20 {
\b\fs32 Introduction to XML}
\par \pard \li720\sb120\sa120\widctlpar {\fs24\ul Mr.
Black} {\fs24  has reviewed the topic list you supplied
and he believes the topics you suggested may be covered
in four seven-hour days. At your option these could be
either four contiguous days or two days in one week and
two days the following week. The basic outline we suggest
would be: }{\b\fs24 \par }\pard
\fi720\li720\sb120\sa120\widctlpar {\b\fs24 1.
Introduction to the basic concepts of XML}{\fs28  }
{\fs24 \par }\pard \li720\widctlpar
{\fs24 This introduction will cover the basic principles
and syntax of XML.........
```

Weaknesses of format-based markup

- Changing the look requires changing codes *in the data*
- Two different looks require two sets of data
- Poor retrieval precision
- No semantic information
- Things that *are* the same may not be *tagged* the same way
  - Different contents, same code
  - Similar contents, different codes

Things that look the same may not be the same


a section header
"emphatic" text
a cross-reference
title of a different work

Widget Maintenance

When operating the Spiffy Widget, it is very important to maintain proper fitness in its casing. (*See Figure 1.*) *In no circumstances* should you operate a loose widget. Consult the *Spiffy Widget Maintenance Manual* for instructions on tightening the widget, or return it to your local Spiffy distributor for maintenance.

What's wrong with format-based markup?

- Interchanges badly
  (designed for one output medium in one application)
- Formatting may be broken when viewed on other platforms/packages
- Vendor-dependence means
  - One vendor only
  - One or two software packages
    - Hard to search and manipulate the information intelligently
    - Information reuse is difficult
  - We are hostage to the upgrade cycle

The alternative: generic markup

- Design your format for the kind of document
  not for one software package's processing
- Identify distinct pieces of a document by name
  - title, paragraph, conclusion

  not appearance
  - font = helvetica, size = 16 pt, weight = bold
- Appearance of a document is *not described*:
  so logic of the content is separated from its appearance

Mr. Black's example

(See slide 10)

```
<article><title>Introduction to XML</title>
<para><person emp.no="BL432">Mr. Black</person> has
reviewed the topic list you supplied and he believes the
topics you suggested may be covered in four seven-hour
days. At your option these could be either four contiguous
days or two days in one week and two days the following
week. The basic outline we suggest would be:
<outline><segment><head>Day One</head>
<title>Introduction to the basic concepts of XML</title>
<para>This introduction will cover the basic principles and
syntax of XML; it will allow those students unfamiliar
with XML to learn the necessary concepts and vocabulary,
and refresh the subject for those students already
familiar with XML.</para>
<para>Situations in which XML is and isn't appropriate
will be discussed,...</para>...
</segment>...</outline>...</para>...</article>
```

Why is generic markup better?

- Allows more than one appearance/application for the same data
- Can be interchanged among machines, operating systems, and software packages without re-keying or reformatting
- Can have entirely different systems for
    - Document creation
    - Document management
    - Document presentation/publication

  working on *the same data*
- “Future-proof”: current text will work on future systems (with a standard syntax)

Production advantages of generic markup

- Improved document management
  (generic markup creates a “self-describing ” textbase)
- Style can be consistent
- Standards can be enforced more easily
- Interchange is improved
- Production process is improved

Disadvantages of generic markup

- Usually does not work “out of the box”:
  requires design, infrastructure
- Therefore does not work as well on small scale
- Design must “fit” to pay off

Types of markup (functions)

- Content of the data
- Structure of the document
- Value-added information
    - Location and navigation
    - Metadata
- When necessary, rendering/processing information
  (presentation, formatting, behavior)

A real-life example

```
<div1 id="sec-intro">
<head>Introduction</head>
<p>Extensible Markup Language, abbreviated XML, describes a class
of data objects called <termref def="dt-xml-doc">XML
documents</termref> and partially describes the behavior of
computer programs which process them. XML is an application
profile or restricted form of SGML, the Standard Generalized
Markup Language <bibref ref="ISO8879"/>. By construction, XML
documents are conforming SGML documents.</p>
<p>XML documents are made up of storage units called <termref
def="dt-entity">entities</termref>, which contain either parsed
or unparsed data. Parsed data is made up of <termref def="dt-
character">characters</termref>, some of which form <termref
def="dt-chardata">character data</termref>, and some of which
form <termref def="dt-markup">markup</termref>. Markup encodes a
description of the document's storage layout and logical
structure. XML provides a mechanism to impose constraints on the
storage layout and logical structure.</p>
<p><termdef id="dt-xml-proc" term="XML Processor">A software
module called an <term>XML processor</term> is used to read XML
documents and provide access to their content and
structure.</termdef> <termdef id="dt-app" term="Application">It
is assumed that an XML processor is doing its work on behalf of
another module, called the <term>application</term>.</termdef>
This specification describes the required behavior of an XML
processor in terms of how it must read XML data and the
information it must provide to the application.</p>
...<div1>
```

A little history

Generic markup is nothing new

- Systems have been using generic markup since 1960s
- A standard syntax for markup languages emerged in the 1980s as SGML (ISO 8879:1986) — Standard Generalized Markup Language
    - Widely accepted in industries including publishing, defense, academic research

Still there was no “killer app”...

- SGML widely used, but expensive to develop
- “The Billion-dollar $ecret” (Chet Ensign)
- Only big (or unusually adept) players could benefit from its economies of scale (and they didn't share)

...Until the web

- HTML is an SGML application
- Demonstrated power of a non-proprietary markup standard

So why XML?

- Problems with HTML:
    - Oriented to presentation (web pages)
    - Not suitable for other applications, cross-media
    - No semantic information
- XML designed as an SGML subset that's easy to develop on the web
    - Target the “sweet spot” between functional and lightweight
    - Use a small set of rules

So ... *what* is XML?

The word “XML” is used to mean:

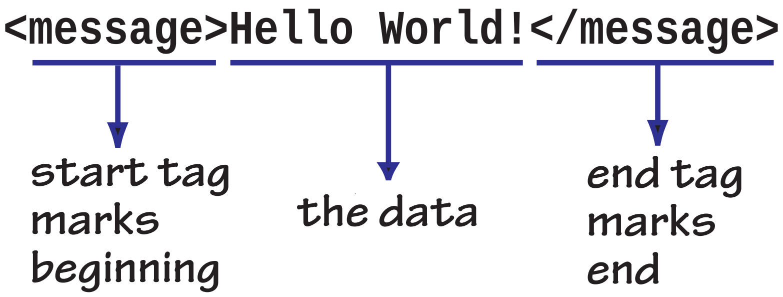
- An open standard (well... a W3C Recommendation) that provides:
    - A data format (standard syntax for markup languages)
    - A data modeling language
- The use of XML-formatted data in an application (like a browser)
- A metalanguage for creating markup languages
- A set of associated recommendations and specifications

    (link, style, query, language specification, transformation, APIs, etc.)

XML works through tags

Paired tags:

- Enclose data
- Identify/name the data
- Named component called an “element”



`<message>Hello World!</message>`

start tag marks beginning

the data

end tag marks end

XML documents

- In XML jargon, your data (no matter what form) is called a “document”
- A document is a coherent, ordered collection of information:
  - invoice
  - journal article
  - topic in a help system
  - database load file
  - reference book

How XML looks at data

Documents

- are made up of *Elements*
- consisting of *Markup* (“tags”)
- ... and *Element content*

A standard syntax

- If a marked up document (data set) follows XML rules, it is called *well-formed*
- All XML is well-formed by definition
  Not well-formed = not XML

Rules for well-formed XML

- *Tags* are delimited by angle brackets “<” and “>”
- Tags delimit *elements*
- Start tags and end tags are always required:

```
<title>Frankenstein</title>
```

- Elements can contain other elements, but may not overlap:

```
<cite>
  <title>Frankenstein</title>, by
  <author>Mary Shelley</author>. 1831 ed.</cite>
```

A couple more rules

- Information can be associated with elements on *attributes*:
  `<title` **`ed="1831"`**`>Frankenstein</title>`
- *entity references* (labels referring to things not in line) are delimited by “&” and “`;`”
  e.g. **`&copy;`** for ©
- Also there are rules about what can be an element or attribute name (e.g. no spaces), how to make comments, etc.

Benefits of a small, strict set of rules

- Easy to recognize when they've been broken
- Easy to know what to do when broken (fix or throw away)
- Easy to build and test tools

What the rules don't say

- What your data is like
- How it's put it together
- What you're going to do with it
- How you're going to do it

Significance of standard syntax

A standard syntax means a standard data model

- Sets of tags may be different
- But all XML languages can:
    - Share parsers, processors
    - Move between systems and behave predictably
- Standard format provides medium for transmission and interchange

Works for any kind of structured data

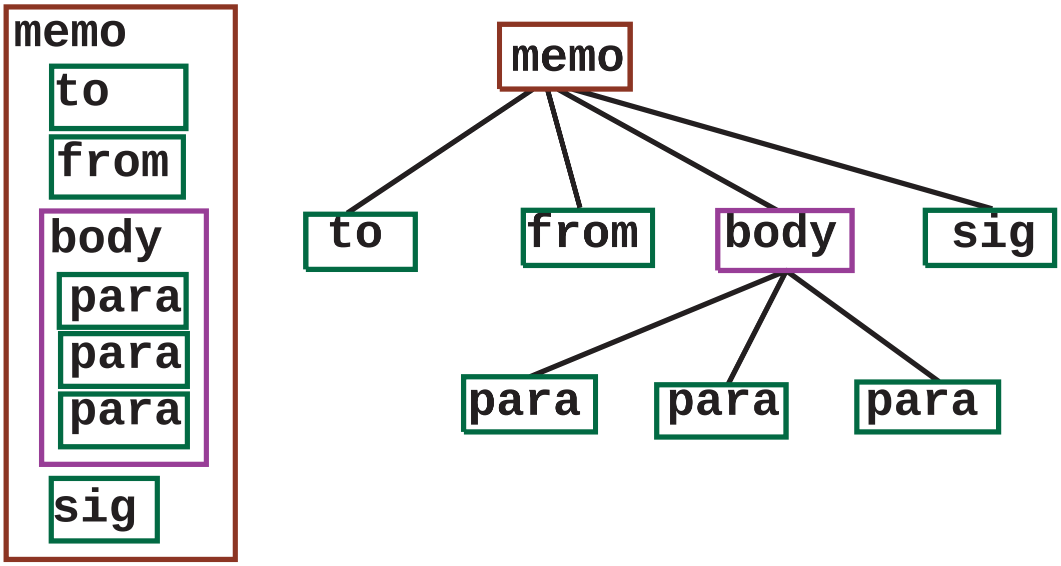
An XML-based information model can be applied to

- Database record sets
- Transaction data, system configurations, code....
- Documentary information (“semi-structured data”)

XML makes nested structures

- Tags identify the start and end of each structure, *not* simply the start of a format
- An invoice might contain:
  - *Tracking number* followed by
  - *Billing code* followed by
  - *Shipping address*, containing
    - *Recipient name* followed by
    - *Street address* (containing...) ... followed by
    - *Contact phone number* followed by…

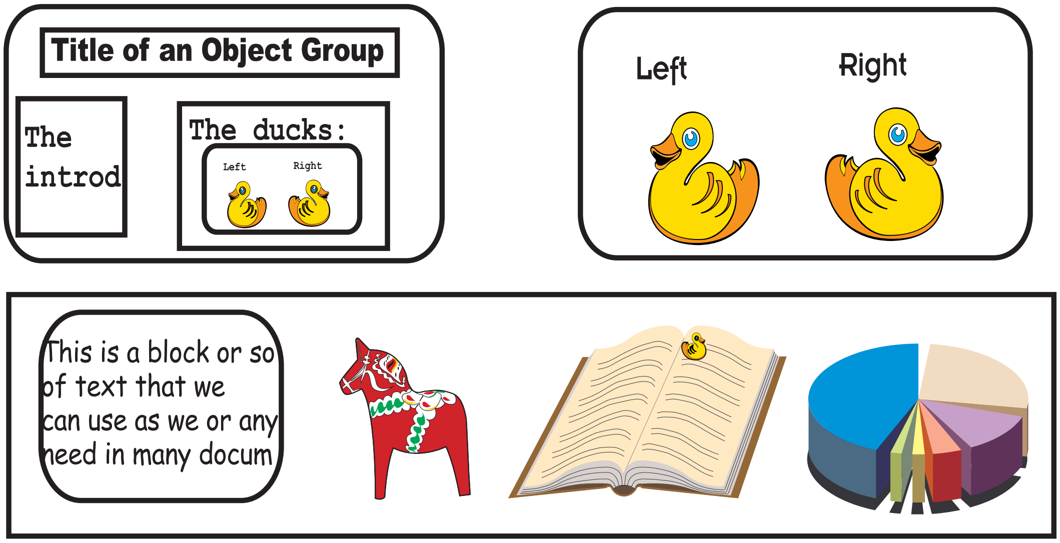
Element nesting = tree structure



---

slide 39

Structured documents contain nested, retrievable objects


Title of an Object Group
The introd
The ducks:
Left
Right
Left
Right
This is a block or so of text that we can use as we or any need in many docum

---

slide 40

Advantages of structured documents

- Store, retrieve, and reuse objects
- Limit a search to within an object
  (or exclude an object from a search)
- Handle at any level of granularity (detail)
- Automatically derive access materials from named structures
  (table of contents; packing list;
  list of names, places, part numbers, etc.)

slide 41

In XML, structure comes for free

Strict rules about required end tags and nesting mean…

- XML documents are structured
- All XML processors see the same structure
  (do not need to *infer* it)
- …organization is built into format, not provided by application

What can you do with data like this?

slide 42

There are many ways to process XML

Options include

- Use a script in Perl, Javascript, Python, OmniMark or other
- Use a program in Java, C, C++, VB
- Filter into a database
- Route into a messaging system
- Display on PDA, Electronic billboard, refrigerator....
- Use XML directly to drive a device
- Make HTML for a browser
- Use XML in the browser
  (HTML not required)
- Use a stylesheet and transformation engine

---

slide 43

XML tagged data for information interchange

- Among business partners (E-Business transactions B2B, B2C, EDI replacing EDIFACT or proprietary formats)
- By data aggregators (semiconductor industry, aircraft)
- Through the life-cycle of a product (among divisions)
- Direct machine-to-machine transfer
- Between rival proprietary formats
- Inter-process communication (IPC)

---

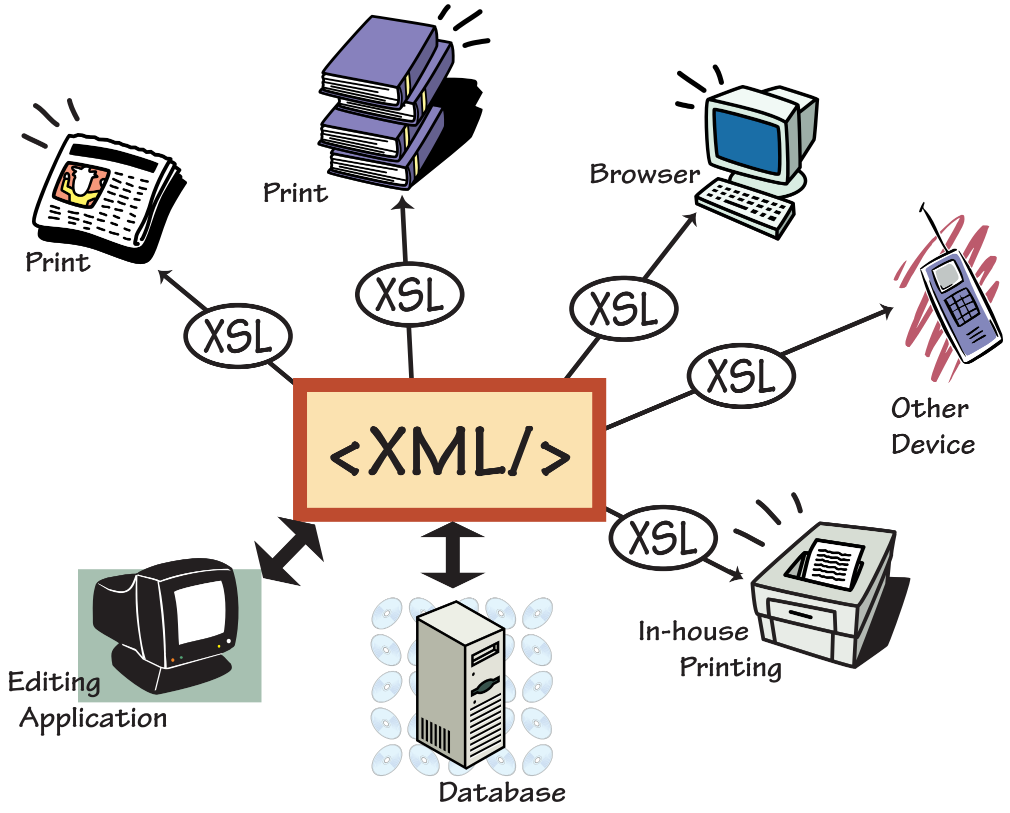
slide 44

XML in a repository

- Content management at many levels of granularity
- Combine data form many sources
- Reuse and repurposing of data / Electronic slice and dice data
- Increase searching precision
- Feed many applications from one repository
- Customized output
- Enterprise information portals

Output in different formats from a single source

Sometimes called “cross-media” publishing



Use stylesheets (such as XSL) to convert a single source into multiple output formats.

XML for publishing

- Many different outputs, one manageable source
    - Many media/device types
      (Web, CD-ROM, handheld PDAs, voice-synthesis)
    - Many styles of print/display
- Different hardware, software, OS for input, manipulation, display
- Publish on demand/Customized output

XML for metadata

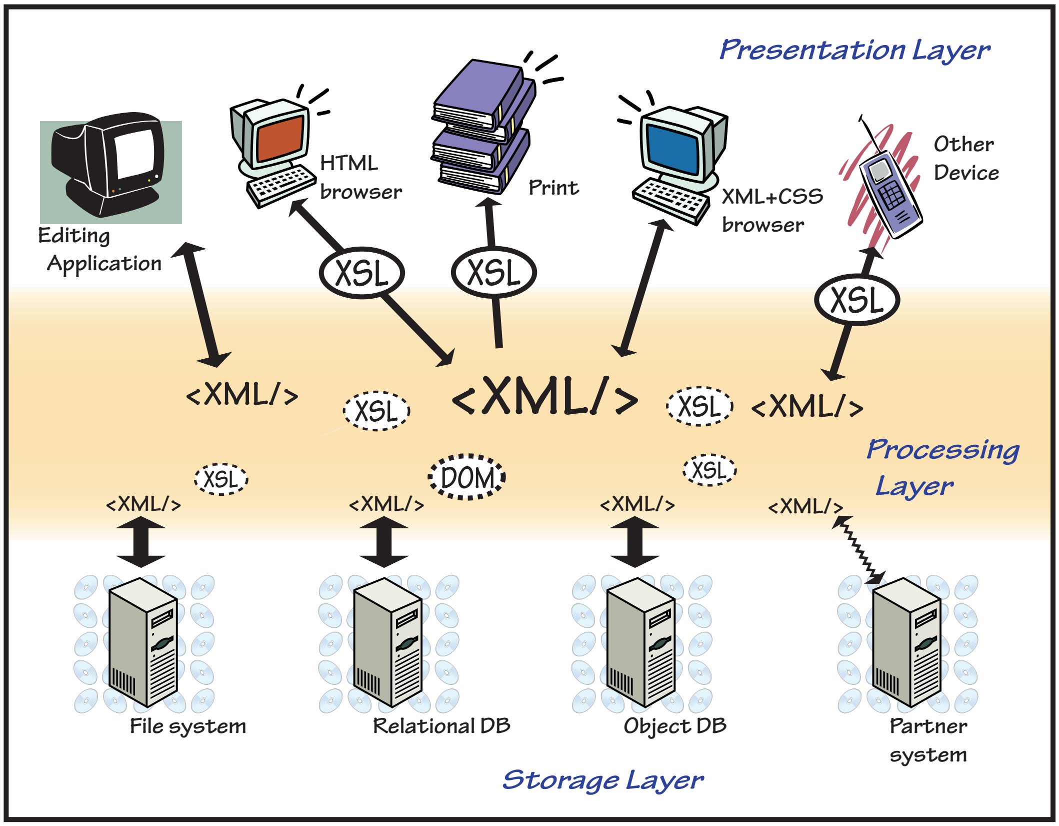
(metadata is information *about* the data)

- Data can remain in a proprietary system;
  separate XML records contain metadata
- Metadata part of an XML data system
- For example:
    - Application metadata (UML models, specifications, requirements)
    - Properties of user interfaces
    - Security levels, access, authority
    - Version control, configuration management, tracking
    - Bibliographic / cataloging data for formal publishing

XML between application layers

- In the “Three-tier” system model:
    - User interface layer
    - Processing or “business logic” layer
    - Storage or repository layer
- XML used in any of the three tiers, *especially in the middle*
- XSL is used for any processing
    - Within the middle tier, and
    - *Between* tiers

A Typical “3-tiered” Model



XML integrates services on a web base

- Web-based “data-centric” integration
    - Allows loose coupling of applications: fault-tolerant and open
- Web becomes friendly to other media
    - Information syndication, on-demand publication
    - Web and print services complementary
    - Information into and out of databases
- From existing data systems directly to the web

The genie is out of the bottle

- Standard markup language syntax creates commodity market for tools
  (like socket wrenches for nuts and bolts, or gasoline for engines)
- Moving data across applications and platforms much easier
- Not just useful for documents...what's a “document”?
    - Purchase orders
    - Claims records, patient records
    - Merchandise catalogs
    - Inventories, manifests

One consequence

Data becomes dynamic

- Markup languages traditionally (still) a way to create and manage content
- But sometimes content drives process
  - e.g. `<chargeCardNo>####-####-####-####</chargeCardNo>`
- XML becomes a *lingua franca* for machine-machine communication

Data modeling in XML

This power comes at a price: *how do we manage it?*

In particular: how can *your* XML work in *my* system?

- We need not just common syntax
- We need to know we “mean” the same thing with our markup (*semantics*)

Remember what XML is...

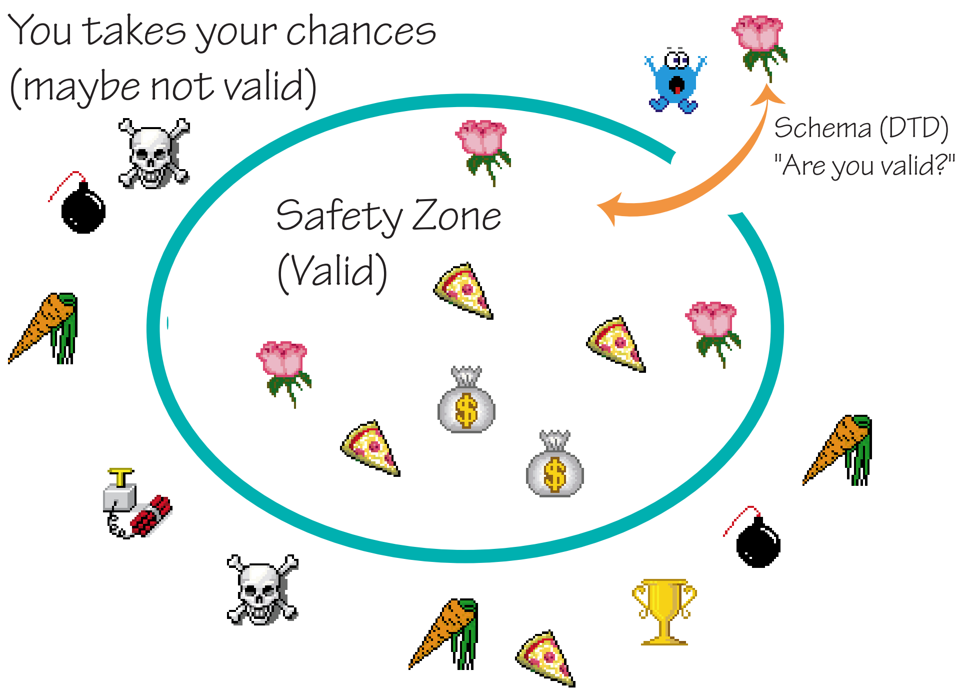
...A standard syntax *and a modeling language*

- XML 1.0 defines a language for a *Document Type Definition* (DTD)
  - Syntax based on SGML DTD syntax
  - Describes a “type” (class) of similar documents
- A DTD is an example of a kind of *schema*
- Allows *validation* of documents
- Makes processing more predictable

Two kinds of validation

- Structural validation
  - “A chapter may contain paragraphs”
  - “But a paragraph may not contain any chapters”
  - Traditional in markup technologies
- Data type (content) validation
  - “Is `2001-02-31` a date?”
  - Common in database technologies
- XML DTDs do structural validation, not data type validation
- Other techniques and languages (not all standard) can help with data type validation

Validation helps you create a safety zone



A caveat: “unsafe” doesn't mean a security threat (no viruses here). They just won't *work* for you.

How an XML DTD is used

- To enforce and document constraints in a markup language
- Can be the basis of a formal contract
- Can specify (automate) application development and configuration

An XML DTD is one type of *schema*

Other schema technologies are emerging

- XML Schema (W3C)
    - Datatype validation, inheritance, element typing
- RELAX (ISO)
    - Easy syntax, clear modeling
- XML-Data (MS)
    - Support in Microsoft tools
- Many other approaches
    - Stylesheet-based analysis and validation, etc....

XML creates markets

- For software
- For services
- For information
- For standards

A market for software applications

- Parsers, processors, component libraries
- Integrated Development Environments (IDEs)
- Editors, content creation systems
- Content and document management solutions
- Repository and textbase technologies

A market for services

- Integration
- Design
- Conversion
- Transformation, analysis, presentation, representation
- Archiving
- Linking, data enhancement, portals....

A market for information

Since XML is an *enabling* technology...

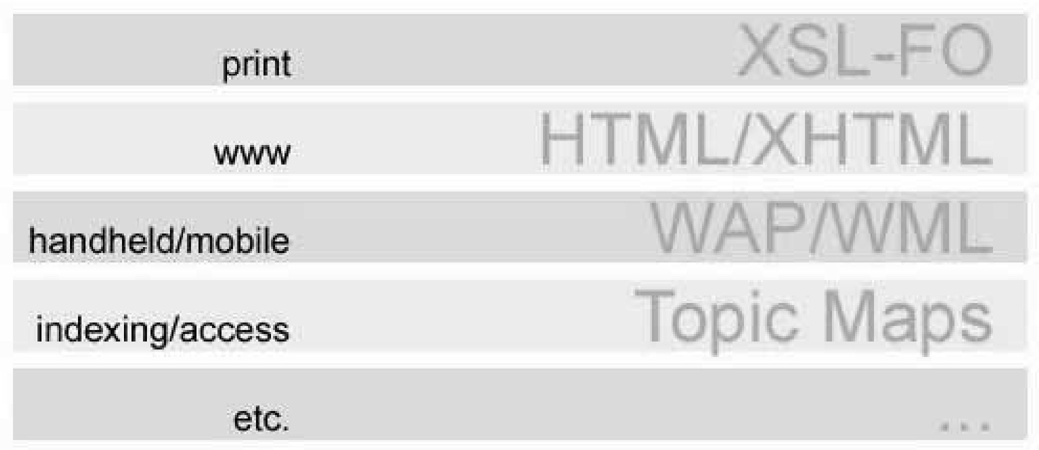
- There will be (is) a market for information *about* XML, and
- There will be (is) a market for information *in* XML, *and* —
- XML disappears below the surface

A market for standards

- XML for media types and applications: “horizontal markets”
  - Presentational or functional markup
- XML for application domains: “vertical markets”
  - Generic markup
- XML technologies: specifications for how to validate, process, apply XML

XML for applications

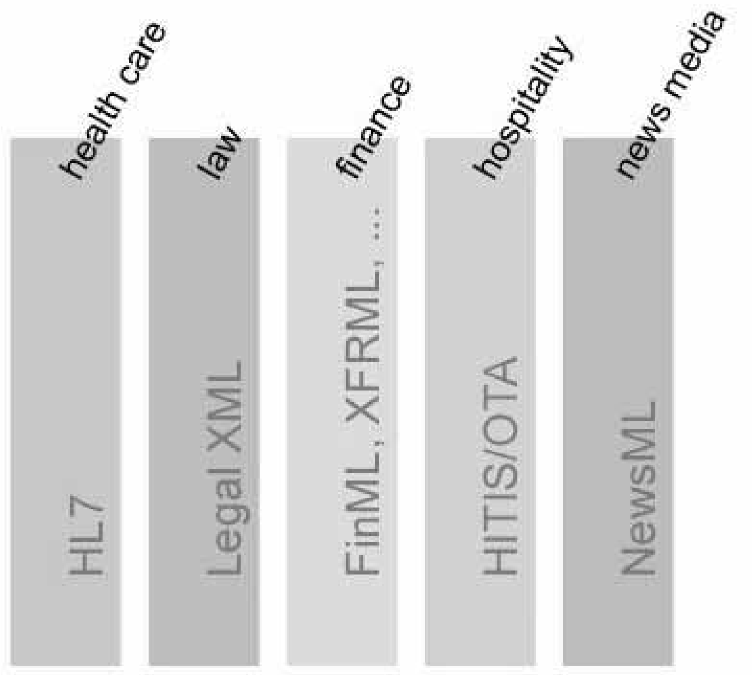
Including various media types



print	XSL-FO
www	HTML/XHTML
handheld/mobile	WAP/WML
indexing/access	Topic Maps
etc.	. . .

Different (presentational) markup languages support different “target” media

XML for application domains



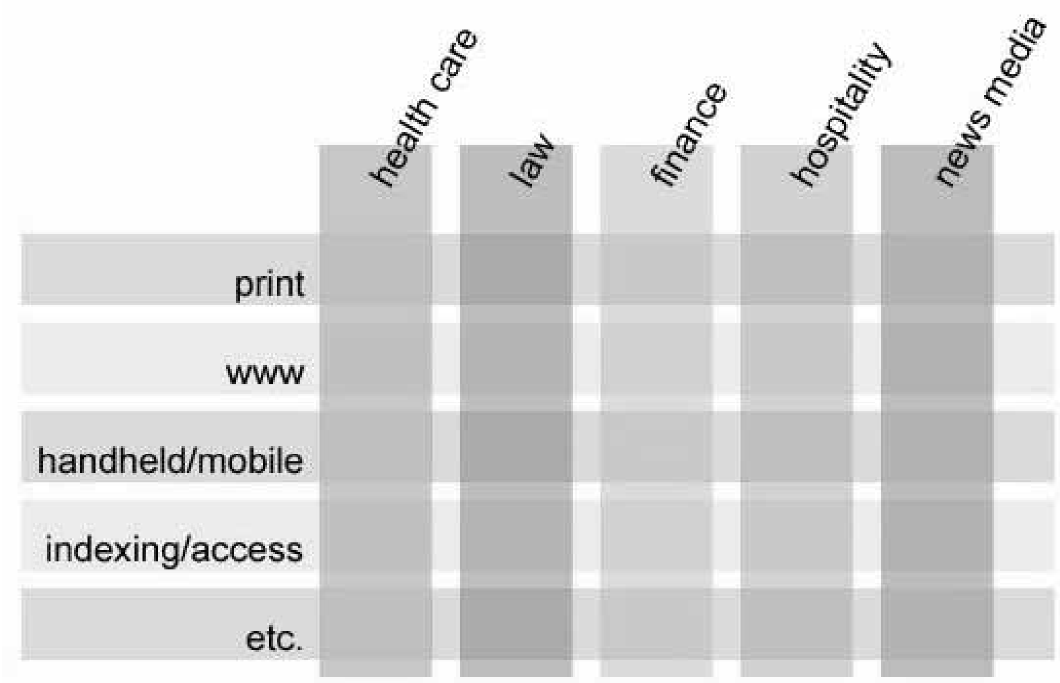
Different (generic) markup languages support different problem domains / vertical markets

Just for example....

Hospitality Industry Technology Integration Standards (HITIS) at **`http://www.hitis.org/`**:

```
...
<SpecialRequests>
  <SpecialRequest ReservationActionType="New">
    <SpecialRequestRPH/>
    <RequestCode>ret222</RequestCode>
    <RequestComments>need wheelchair for handicap
guest</RequestComments>
  </SpecialRequest>
</SpecialRequests>
<Comments>
  <ProfileComment>
    <CommentOriginatorCode>comm123</CommentOriginatorCode>
    <GuestViewable>yes</GuestViewable>
    <LastUpdated>1999-11-10T10:23:47</LastUpdated>
    <Comment>additional comments about this
profile</Comment>
  </ProfileComment>
</Comments> ...
```

The XML application grid

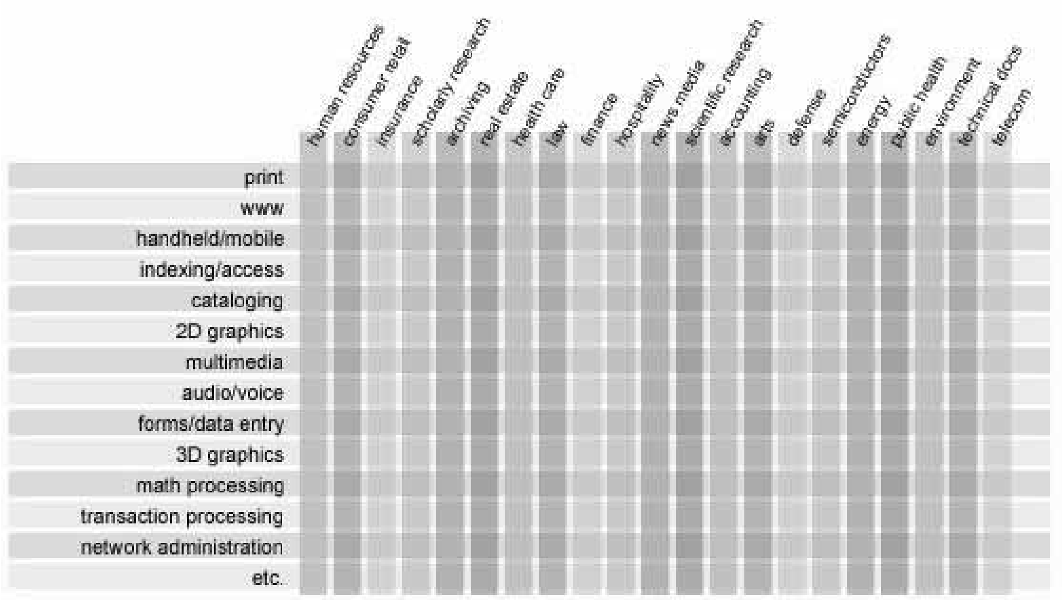


Intersections are opportunities for (automated) data *transformation*

---

slide 68

The grid, v. 2



One design can't solve all problems: sensible scoping is essential

---

slide 69

How do you deal with alphabet soup?

- Understand the basics
- Understand your problem and break it down
- Understand what XML 1.0 is — and is not
- Shine a light before you go down a blind alley
- Go *in*, then go *out*
  - First, understand your information
  - Then, decide what you want to do with it
  - *Then*, research options and opportunities

The XML family of standards

XML Schema

Full-featured schema language

XSLT

Stylesheet/transformation language

XLink/XPointer

Hypertext linking

DOM, SAX

APIs for access and control of XML “documents”

RDF

Cataloging / resource management

etc.

XML Schema

- Full-fledged schema language
- Both structural and datatype validation
- In XML syntax
  - One parser for document and its schema
  - Documentation-friendly
- Current status (Nov. 1 2000): W3C *Candidate Recommendation*

An example of XML Schema

```
<xsd:element name="PARA">
  <xsd:complexType mixed="true">
    <xsd:sequence>
      <xsd:element name="FOOTNOTE">
        <xsd:complexType mixed="true">
          <xsd:sequence>
            <xsd:element name="BIBL" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="I" type="xsd:string"/>
      <xsd:element name="B" type="xsd:string"/>
      <xsd:element name="DATE" type="xsd:date"
minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

DTD vs. XML Schema?

- XML 1.0 DTD syntax works now
  - Arcane syntax
  - Structural validation
  - But well understood
- W3 XML Schema should be ready soon
  - More versatile, with more complex kinds of validation than DTDs
  - May take some time to mature (both tools and best practices)
  - Easier or harder than DTDs? the jury is still out

slide 74

Other validation techniques

Many different choices:

- Alternative syntaxes and proposed syntaxes, e.g. RELAX, DCD, DDML, XML-Data ...
- Stylesheet-based approaches, e.g. Schematron
- Validation can be hard-wired into applications
- All have their strengths; mix and match!

slide 75

XSL and XSLT

- XSL: full-featured stylesheet language for defining presentations of XML in other forms
- In two parts:
- XSL-FO — XSL formatting objects
- XSLT — XSL transformations

XSL Formatting Objects

- Abstract specification of formatting objects for print, web, alternative media output
- Supports complex layout design: right-to-left, top-to-bottom writing; arbitrary layouts in screen or page sequences and regions, running heads, floats, etc. etc.
- Allows device-independent, complex presentation of any XML
- Versatile architecture: processing either on client or server
- Current status (Nov. 1 2000): W3C *Working Draft* (been to Last Call once)

XSLT: Transformation Language

- Language to specify transformations or mappings of XML into other formats
- Designed for XSL-FO, but useful for other conversions
- Commonest use: create HTML from XML source for the web
- Other uses: XML to XML; XML to plain text; graphic rendition of XML (SVG); generate reports for analysis/QA/validation; etc. etc.
- Current status: W3C *Recommendation* since November 1999

XLink and XPointer

- XLink: specification for linking in XML
    - XML syntax is great ...
    - ... but if there's no linking, there's no web!
- XPointer: how to specify a link when you don't have a name for the target

XLink: basic linking in XML

- Any element can be a link
- Links from any XML to any XML and non-XML too
- Two-way links, multi-ended links, links with special behaviors
    - Rendition by type
    - Can bind to authentication/verification protocols
    - Can have fallback behavior
    - etc.
- Can be maintained in separate documents (“link sheets”)
- Current status: W3C *Candidate Recommendation* since July 2000

XPointer: addressing *into* documents

- Built on XPath, an addressing syntax
    - e.g. `/PLAY/ACT[3]//LINE[contains(., 'tomorrow']`
    - Also part of XSLT
- Link to documents with stable structure, dynamic content
- Link to public documents you do not control
- Basically a query language

More related standards

- DOM (W3C): Document Object Model, an object-oriented API
- SAX (independent): Simple API for XML, a basic event-based parser API for Java and other languages
- RDF (W3C): Resource Description Framework, a protocol for describing relationships between data objects
- Topic Maps (ISO): a protocol for describing and labelling high-level relationships to drive indexing and access
- others?

For more information

The source for XML and related topics

- Robin Cover's SGML/XML Web Page: **`http://www.oasis-open.org/cover/sgml-xml.html`**

General XML information

- W3C's XML page: **`http://www.w3.org/XML/`**
- XML FAQ (Peter Flynn): **`http://www.ucc.ie/xml/`**
- XML.com: **`http://www.xml.com`** (industry coverage and tools)
- XMLinfo.org **`http://www.xmlinfo.org`** (covers tools and development)
- XSLinfo.org: **`http://www.xslinfo.org`** (covers XSL development and implementation issues)

XML e-business and EDI information

- The OASIS and UN/CEFACT entry**`http://www.ebxml.org`**
- CommerceOne's entry**`http://www.commerceone.com`** (Common Business Library etc.)
- The Microsoft entry**`http://www.BizTalk.org`**
- RosettaNet consortium **`http://www.rosettanet.org`** (vocabulary and process issues, industry coverage)
- XML.ORG **`http://www.xml.org`**
- Commerce XML (cXML)**`http://www.cxml.org`**
- Oracle's XML material **`http://www.oracle.com/xml/content.html`**
- IBM is heavily into this, too.**`http://www.developer.ibm.com`** [for Business Rules Markup Language (BRML) and Trading Partner Agreement Markup Language (tpaML)

And others too numerous to mention: CommerceNet's eCoFramework, SAIC's Universal Commerce Language and Protocol (ULCP), XEDI.org, etc.

Printed books on concepts

- **SGML: the Billion-Dollar Secret**, by Chet Ensign (Prentice-Hall PTR, 1997)
    - Manager level. Written about SGML (XML's parent standard), but almost entirely applicable: excellent on issues of scalable system development.
- **ABCD… SGML**, by Liora Alschuler (Thompson Computer Press, 1995)
    - Written about SGML (XML's parent standard), but change the word "SGML" to "XML" as you read it and it still applies. Talks about work process changes an XML system can bring.
- **XML: A Manager's Guide**, by Kevin Dick (Addison-Wesley Information Technology Series, 2000)
    - Manager level. Solid view, but stays at 10,000 feet up.
- **The XML Companion (2nd Edition)**, by Neil Bradley (Addison-Wesley, 2000)
    - Very good basic technical introduction.
- **Professional XML**, by Richard Anderson, Mark Birbeck and ten more authors. (Wrox Press Ltd.)
    - Light technical level. Each author wrote an introduction and then examples/case study for one technical topic. Introduces the problems of XML and databases, the XML APIs DOM and SAX, server to server XML (XML-RPC, SOAP, etc.) and more.

Other information sources

- **Markup Languages: Theory and Practice** (a quarterly journal): **`http://mitpress.mit.edu/MLANG`**
- OASIS Home Page (vendor consortium): **`http://www.oasis-open.org`**
  - XML.ORG: **`http://xml.org`** (document model repository and support materials)
  - OASIS XML Conformance Subcommittee: **`http://www.oasis-open.org/committees/xmlconf-pub.html`**
- Graphic Communications Association: **`http://www.gca.org`** (sponsors conferences including XML Europe, Paris, June 2000)
- XML.COM: **`http://www.xml.com`**

Still more information sources

- Basic newsgroup: `comp.text.xml` (also some on `comp.text.sgml`)
- Useful Lists
  - XML-L: **`http:/listserv.heaanet.ie/xml-l.html`** (for newcomers)
  - XML-Developer's List: **`http://www.lists.ic.ac.uk/hypermail/xml-dev`** (heavy technical discussion)
  - XSL-List: **`http://www.mulberrytech.com`**