Why XML for Print?

B. Tommie Usdin
Mulberry Technologies Inc.
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 (January 2006)

Mulberry
Technologies, Inc.

Why XML for Print?

slide 1

Why XML for Print?

- Print is only one of many uses (maybe primary, maybe not)
- Print is not just about *one* product
- Vendor-specific files are a trap
- Processing gains
  - single source
  - quality
  - speed
- XSL-FO, print directly from XML
- Busting some myths concerning XML and print

slide 2

Print is Only One of Many Uses

slide 3

“Print” Now Encompasses More

“We consider our online delivery part of our print business”

Elsevier executive (5 years ago!)

slide 4

XML is All About Repurposing and Reuse

(The“Why XML” Clichés)

- Print is not enough any more
- Single-use data is too expensive
  - Information is a corporate resource and must be managed accordingly
  - If we can’t get our data out, we don’t want it in
- Standard pages aren’t enough anymore; every client needs a personalized look and feel
- I want just-in-time merge from *this* form plus *that* database
- Web design and print design are different

slide 5

Companies are Using XML for Print

When the Same Content Must Be:

- Printed, as we’ve always done
- Published on website
  (cross-references and bibliographies as live links)
- Delivered to newsfeeds
- Placed in the data repository
  - reuse / repurpose the same material
  - update continuously
  - maintain digital rights and permissions
- Provided to content aggregators
  (meansbusiness.com, books24x7.com, EBSCO, Mead, JSTOR, PubMed Central, Ithaka)
- Electronically reviewed and revised
- Printed again, in new products or formats

Ultimate Purpose of XML

- Encode (mark up) data only once (not once per product)
- Construct *many* products from that markup
- Reuse data (in whole or part) many times
  - print publications
  - websites and online syndication
  - ebooks and publishing to other devices (PDAs)
  - electronic archives for search and reuse
  - new product opportunities

slide 7

One XML Document and Many Results

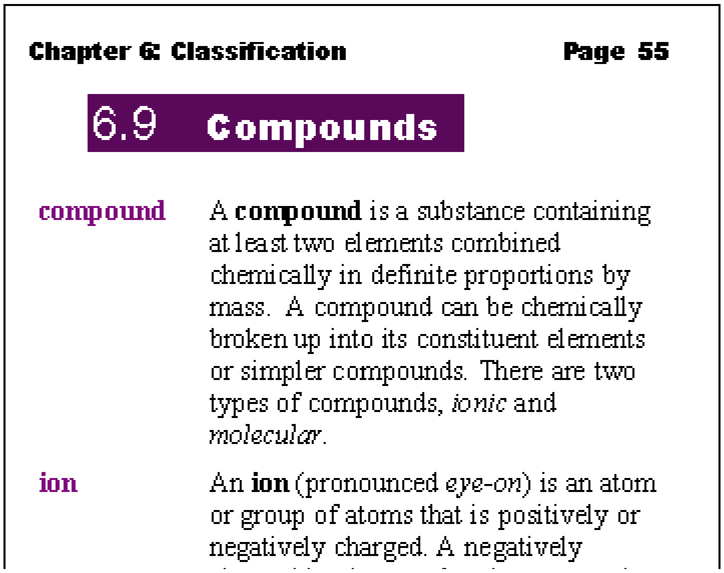
(recombination)

```
<section id="F8493842" lastupdate="2001-05-22">
<title>Compounds</title>
<para>
A <keyterm>compound</keyterm> is a
substance containing at least two elements combined
chemically in definite proportions by mass. A compound
can be chemically broken up into its constituent elements
or simpler compounds. There are two types of compounds,
<term>ionic</term> and <term>molecular</term>.
<question-and-answer>
Testbank <testgroup>GDW</testgroup>
<question-group>
<question>6</question><question>7</question>
<question>9</question><question>54</question>
</question-group>
</question-and-answer>
</para>

<para>An <keyterm>ion</keyterm>
(<pronunc>eye-on</pronunc>) is an atom or group of
atoms that is positively or negatively charged. A
negatively charged ion is an <keyterm>anion</keyterm>
(pronounced <pronunc>an-eye-on</pronunc>) while a
positively charged ion is a <keyterm>cation</keyterm>
(pronounced <pronunc>cat-eye-on</pronunc>). An
<keyterm>ionic compound</keyterm> is a compound that
is held together by the attractive forces between
positively and negatively charged ions.
<question-and-answer>
Testbank <testgroup>GDW</testgroup>
<question-group><question>6</question>
<question>7</question> ionic compounds</question-group>,
<question-group><question>9</question> cations<question-group>.
<question-group><question>25</question>
<question>26</question> anions<question-group>
</question-and-answer>
</para>
```

slide 8

The Print Textbook



Chapter 6: Classification Page 55

6.9 Compounds

compound

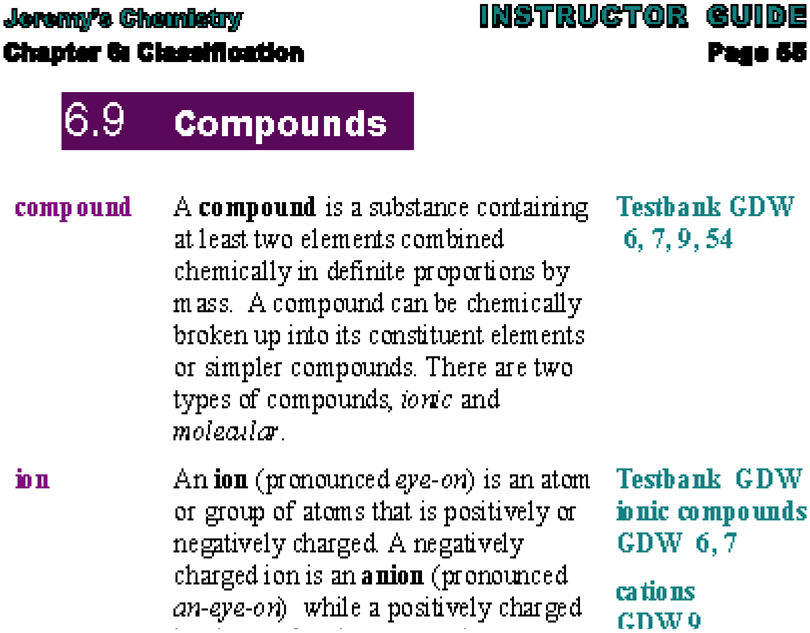
A **compound** is a substance containing at least two elements combined chemically in definite proportions by mass. A compound can be chemically broken up into its constituent elements or simpler compounds. There are two types of compounds, *ionic* and *molecular*.

ion

An **ion** (pronounced *eye-on*) is an atom or group of atoms that is positively or negatively charged. A negatively

slide 9

The Instructor's Manual (Print and Web)



Jeremy's Chemistry INSTRUCTOR GUIDE

Chapter 6: Classification Page 55

6.9 Compounds

compound

A **compound** is a substance containing at least two elements combined chemically in definite proportions by mass. A compound can be chemically broken up into its constituent elements or simpler compounds. There are two types of compounds, *ionic* and *molecular*.

Testbank GDW 6, 7, 9, 54

ion

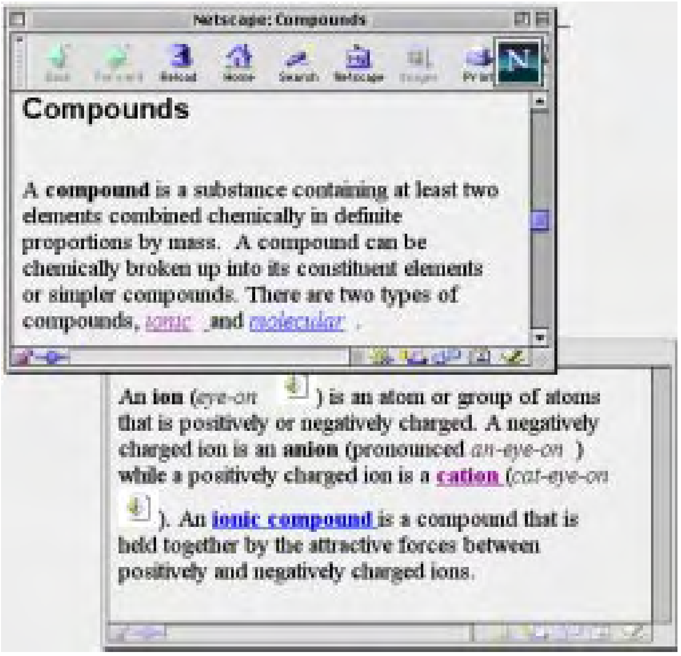
An **ion** (pronounced *eye-on*) is an atom or group of atoms that is positively or negatively charged. A negatively charged ion is an **anion** (pronounced *an-eye-on*) while a positively charged

Testbank GDW ionic compounds GDW 6, 7

cations GDW 9

slide 10

Student Web Page and eBook



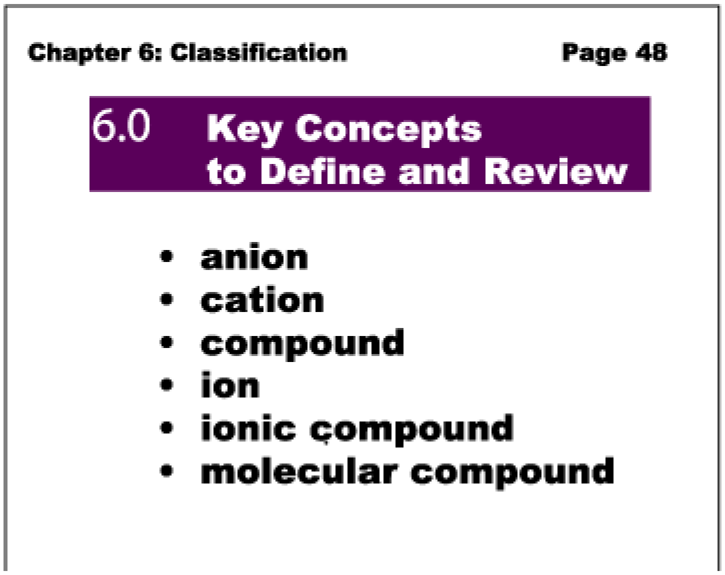
Compounds

A **compound** is a substance containing at least two elements combined chemically in definite proportions by mass. A compound can be chemically broken up into its constituent elements or simpler compounds. There are two types of compounds, ionic and molecular .

An **ion** (*eye-on*) is an atom or group of atoms that is positively or negatively charged. A negatively charged ion is an **anion** (pronounced *an-eye-on*) while a positively charged ion is a **cation** (*cat-eye-on*). An **ionic compound** is a compound that is held together by the attractive forces between positively and negatively charged ions.

slide 11

Automatically Generated for the Same Textbook



Chapter 6: Classification **Page 48**

6.0 Key Concepts to Define and Review

- **anion**
- **cation**
- **compound**
- **ion**
- **ionic compound**
- **molecular compound**

slide 12

In Other Words

- Recombination
- Reuse
- Repurposing
- Slice and dice publishing
- Selection, extraction, and sorting

slide 13

Print is Not Just One Product

- To many folks, print means making one QuarkXPress file
- Taking that file through to 4-color glory
- If that’s a true and you make *one* product, XML is probably overkill
- But *for much of the world* one product simplicity has never been true

slide 14

Variety of Print Publications

- **War Story 1984** — The Boeing Company
  - 8 ½ by 11 Computer Manuals
  - computer small-books for the “new” PC market
  - helicopter and airline pilots manuals
- **War Story 2006** — Major magazine publisher
  - magazines major revenue stream
  - books reuse magazine recipes, boxes, “how-to”s
  - newsletters to specific audiences
  - recipe cards and how-to kits

slide 15

Customization

(change, assemble, or adapt based on customer)

- Mix and match components
- Send each large customer
  - same information
  - packaged and formatted to their specs
- Create a distinct look and feel
  - for a class of users (long-distance runners, corporate executives)
- based on demographics, customer databases

slide 16

Personalization

(tailor a product to an individual person)

- To a particular user (based on purchase history)
- Personalized stock reports
- Personal Wall Street Journal (customer interest-profile)
- Rules-based filtering of material
- Collaborative filtering (your personal preferences and those of similar subscribers)

slide 17

Internationalization

(adapting a product for potential use everywhere)

- Must support
  - multiple languages
  - multiple scripts and writing directions
  - various date-time formats and currency
- Needs flexibility (components that take readily to different design)

War Story: Large manufacturer of consumer electronics

slide 18

Localization

(adapting product look and content to a specific locality/region)

- Must consider
  - cultural sensitivities and values
  - design and aesthetics
  - local dialect as well as language
- May include
  - special local content
  - removing content as well as adding
  - replacement graphics

slide 19

What Else can Print Production Gain from XML Processing?

- Vendor independence
- Simple (powerful) XML transforms
- QA methods and quality gains
- Processing and production gains

Vendor-specific Files are a Trap

slide 20

Most Print Today is Produced Using

- Page layout / composition software
- Desktop publishing package
- Word processor
- Graphics program

slide 21

Vendor-specific Formats Belong to the Vendor

- Embed “codes” in the data to create formatting
- Codes only work with that vendor’s software
- Such codes:
  - *can be changed by the vendor at any time.*
  - can’t be changed by you

slide 22

Vendor-specific Codes May Make it Difficult to

- Extract data in re-usable form
- Make global style changes
- Produce multiple output products (not just pages)
- Change your material to use it in new ways
  - rearrange
  - extract
  - reuse and repurpose data
- Turn what-you-have into another-product
  - QuarkXPress into MS Word
  - XYvision into HTML
  - brochure into course catalog

slide 23

XML Provides True Vendor Independence

- No hardware platform dependencies
- No software dependencies
- No proprietary embedded codes to get rid of
- Change look and feel without recoding
- Make many publications without recoding

slide 24

Part of XML’s Independence is XSLT

(Extensible Stylesheet Language Transformation)

- Transformation and manipulation functions for XML files
- A programming language (of a sort)
- Transforms *from* XML *into* something else
- Is one *really good* mechanism for the rearrangement and reuse

slide 25

XSLT Reads XML Documents and Writes

- HTML for browsers
- XML in a different tag set
- typesetting driver files (InDesign, QuarkXPress, FrameMaker)
- interchange files (RTF, EDI, etc.)
- flat text files (plain text, comma separated, ASCII, etc.)

slide 26

What Organizations Do with XSLT

- Simple business transforms
- Making HTML from richer XML
- **Single Source and Reuse Publishing**
- Transforms for editorial QA
- XML to XML transforms
- XSLT as the middle component in XSL-FO

slide 27

Example: Making HTML From Richer XML

Read in semantically rich XML tagging

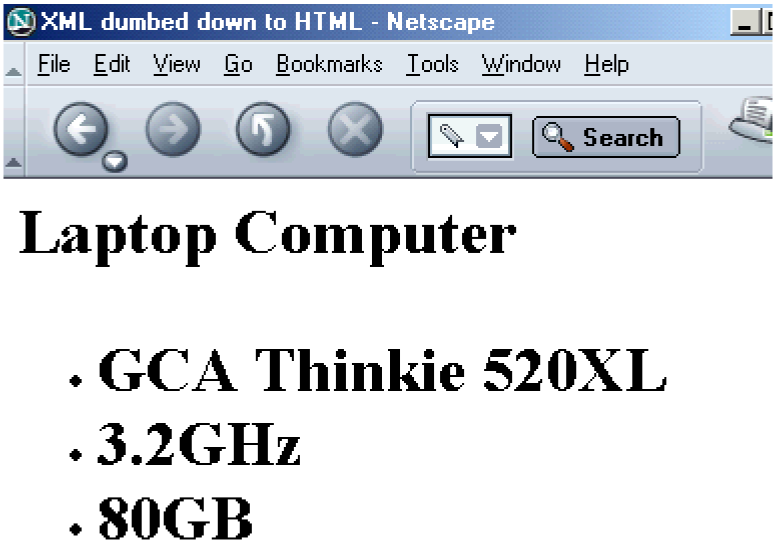
```
<COMPUTER CLASS="Portable">
<MFR>GCA</MFR><FAMILY>Laptop</FAMILY>
<SPEED UOM="GHz">3.2</SPEED>
<LINE>Thinkie</LINE><MODEL>520XL</MODEL>
<DISK UOM="GB">80</DISK>
</COMPUTER>
```

Simplify it to HTML for display in any browser

```
<H2>Laptop Computer</H2>
<UL>
<LI>GCA Thinkie 520XL</LI>
<LI>3.2GHz</LI>
<LI>80GB</LI>
</UL>
```

slide 28

Which Displays As


XML dumbed down to HTML - Netscape
File Edit View Go Bookmarks Tools Window Help
Search

Laptop Computer

- **GCA Thinkie 520XL**
- **3.2GHz**
- **80GB**

slide 29

Better Example: XSLT Makes Single Source and Reuse Publishing Happen

(transformations build products)

- Print on Demand (different users = different order or different stuff)
- Select / Extract / List / Omit
  - Pull the metadata to put into the catalog
  - Extract article titles and abstracts for the advertising webpage
  - Extract the CME material for a special site for nurses
  - Collect environmental impact material
  - Publish this report with all the **SECRET** material removed
  - Send just the citations to a link matching service

slide 30

XML Brings Processing and Production Gains

- Processing speed
- Time to market
- Enhanced content quality
- XSL-FO (print directly from XML)

slide 31

Processing Speed

(The XML Cliché)

There is always a

- Faster
- Cheaper
- Easier

way to do any *one* thing than XML

slide 32

XML can be Significantly Faster

- Eliminate parallel creation and update
- Lights-out publishing (e.g., invoices, medical records, catalogs)
- Separates work on format from writing content
- Validation finds surprises early
- Automate tedious and repetitive handwork
- Citation and cross-reference checking
- Automated formatting virtually eliminates “check that every X is formatted as Y”

slide 33

QA and Quality Gains

- Consistency of formatting look and feel
- Content checking for increased quality
- Fewer surprises removes last-minute production glitches
- Semantically meaningful components
(UNIX command, product name, genus-species)
- Generated text (autonumbering, “Figure 3.”, Table of Contents)
- New proofing and checking methods (see next slide)

slide 34

New Proofing and Checking Methods

limited only by imagination and programmers

- Lists of …
- False-color proofs
- Content checking
- Cross-checking

slide 35

List of “Figures” / List of “Tables”

- List any Element
- Determine
  - how many
  - list them for human checking
  - automated authority file checking
- Show location within text

[country (check code), product (check for trademark),
abbreviated journal title (check authority file), price (correct units)]

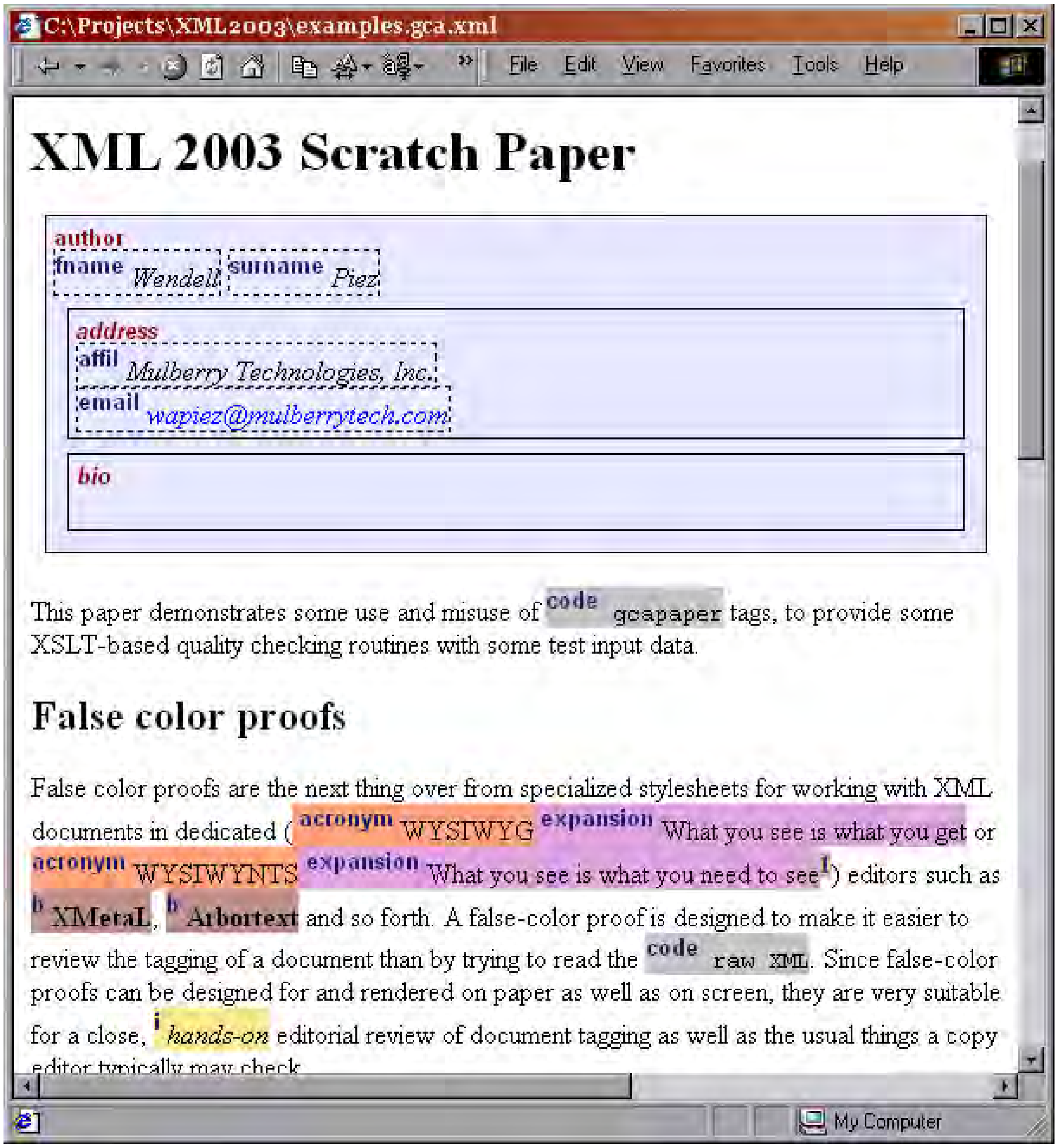
slide 36

False-color Proof

human checking with computer assistance

- “Electronic galley proof”
- Display text in format designed for checking by eye
- Make different elements different colors
- View on screen (HTML) or print
- Especially useful when
  - high-quality “semantic” tagging
  - tag sets subject to tag abuse

An XML 2003 Conference Paper in False-color Proof



XML 2003 Scratch Paper

author
fname *Wendell* surname *Piez*

address
affil *Mulberry Technologies, Inc.*
email *wapiez@mulberrytech.com*

bio

This paper demonstrates some use and misuse of code `gcapaper` tags, to provide some XSLT-based quality checking routines with some test input data.

False color proofs

False color proofs are the next thing over from specialized stylesheets for working with XML documents in dedicated (acronym WYSIWYG expansion What you see is what you get or acronym WYSIWYNTS expansion What you see is what you need to see[1]) editors such as b **XMetaL**, b **Arbortext** and so forth. A false-color proof is designed to make it easier to review the tagging of a document than by trying to read the code `raw XML`. Since false-color proofs can be designed for and rendered on paper as well as on screen, they are very suitable for a close, i *hands-on* editorial review of document tagging as well as the usual things a copy editor typically may check

slide 38

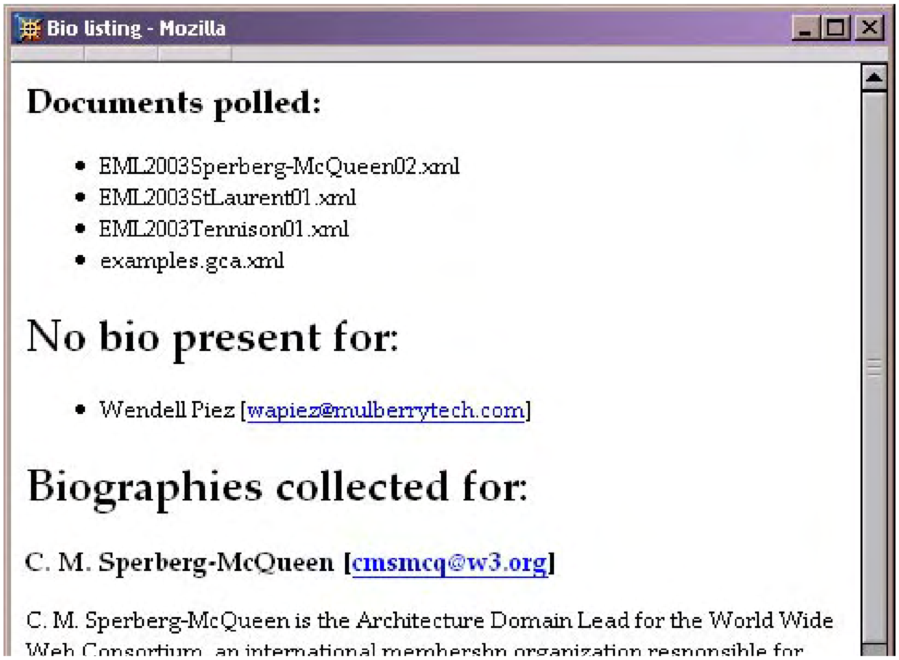
Content and Tag Abuse Checking

- Can work
  - over a single document
  - (more usefully?) over sets of documents
- Check metadata, bibliographies and other semi-structured information
- For example
  - which first-mention `acronyms` lack `expansions`
  - Report for a group of papers
    - authors whose `bio` has no content
    - list the `bio`s you have

slide 39

Example: Filtering for Unfinished Bios

(with links to e-mail the authors)



Documents polled:

- EML2003Sperberg-McQueen02.xml
- EML2003StLaurent01.xml
- EML2003Tennison01.xml
- examples.gca.xml

No bio present for:

- Wendell Piez [wapiez@mulberrytech.com]

Biographies collected for:

C. M. Sperberg-McQueen [cmsmcq@w3.org]

C. M. Sperberg-McQueen is the Architecture Domain Lead for the World Wide Web Consortium, an international membership organization responsible for

See `examples/biolist.html`

slide 40

Two Major Validation Strategies

- Check against XML model (DTD or schema) for
  - missing elements (no title for news article)
  - badly positioned elements (footnotes in the bibliography)
- Test for certain properties of the content
  - locate all codeblocks with lines too long
  - are all figures referenced at least once?
  - do all list items begin with an upper-case letter?
  - do all cross-references point to something?
  - are all part numbers in the database?
  - are dates reasonable? Do death dates follow birth dates?

slide 41

XSL-FO: Format Directly from XML

(For many, a reason to embrace XML)

- Extensible Stylesheet Language-Formatting Objects
- Automated formatting from XML-structured data
- Straight from XML into PDF or Postscript
- Lights-out high-volume production of pages

slide 42

The *Idea* of XSL-FO

- The language defines layout and styles
  - platform-independent
  - vendor-independent

The Dream: high quality, high volume, content-driven publishing
(Not for layout-driven publishing!)

slide 43

How XSL-FO Formatting Works

- XSL provides a tag set into which XML documents may be transformed (using XSLT)
- The tag describe
  - the layout geometry of the page (into which you pour content)
  - a set of *formatting objects*
    - that say how to put content on the page
    - that describe how the document should be rendered
- An XSL-FO *rendering engine* makes pages / display from these tags
- An XSL-FO document is
  - an XML document
  - with text and graphic content wrapped in formatting object tags

slide 44

XSL-FO for Internationalization, Localization, Accessibility

- XML character set is internationalized (Unicode)
- XSL-FO supports non-Western writing direction
  - Left-to-right-top-to-bottom
  - Top-to-bottom-right-to-left
- Set up XSL-FO programs (style specs) once and flow different languages in

slide 45

XSL-FO is a Great Report Writer

(where pagination is not a problem)

- Credit card and bank statements
- Investment portfolios
- Hospital records and patient medical records
- Insurance policies and claims
- State legislatures for bills, resolutions, and reports
- Directory and catalog products

(Anybody where lights out works is a good candidate)

slide 46

In Conclusion Some Myths Concerning XML and Print

slide 47

Canard: XML Doesn’t Need Skilled Designers/Typographers

- They say that XML
  - has stylesheets instead
  - can’t look good anyway
- Truth: XML has no look, but it can take on many looks
- Truth: We need more designers than ever before!
  for different designs for web, print, interactive, personalized, etc.
- Truth: If the presentation is ugly, then we designed ugly.

slide 48

Semi-Myth: Lights Out Production

The Rumor: Formatting based strictly on tags is good enough;
we can do *lights-out* composition
(This last is usually said by composition vendors.)

Truth:

- Works *really well* with short or repetitious data in large quantities
  - invoices, lists, reports
  - medical records, insurance, financial statements
- Works *pretty well* for web publishing (elastic pages)
- *Can* work (with rerunning) for desktop-publishing quality
- Does *not* work for real high-end composition-system quality
- Repeat: Does *not* work for high-end typographic quality

slide 49

Myth: XML is Difficult

(typesetters could never understand it)

Truth:

- XML is enabling
- Service vendors of all stripes are understanding it
- Doing new things with content *is* more difficult,
  don't blame the XML, it can help
- XSL-FO is typography and works lots better with typographers

slide 50

Myth: XML is for Programmers

Truth:

- OK, DTDs and schemas are written by programmers
  (or ought to be)
- Almost anyone can write XSLT transforms
  (programmers write trickier ones)
- Use, re-use, tagging, faster production, and improved QA and quality are for the rest of us

(We made this slide show in XML because that was easier and better)

slide 51

The Bottom Line: Don’t Lock Up My Content

- If the web and all PDAs vanished tomorrow
- If all I ever did was print publication
- I’d still use XML
  - same content with many designs
  - multiple products from one source
  - all the “ations”
    - customiz**ation**
    - internationaliz**ation**
    - personaliz**ation**
    - localiz**ation**

(Almost any business could say this!)

slide 52

Colophon

- Slides and handouts created from single XML source
- Slides projected from HTML which was created from XML using XSLT
- Handouts created from XML:
  - Source XML transformed to Open Office XML
  - Open Office XML opened in Open Office
  - Pagination normally adjusted
  - Saved as PDF
- Slideshow materials available at:
  `http://www.mulberrytech.com/slideshow`