XML and Print

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`

August 2001

Mulberry
Technologies, Inc.

XML and Print

slide 1

Administrivia

- Start, end, break
- How this will work
- Questions are always in order
- Why this course
- Anything else?

slide 2

Where We Are *Not* Going in This Talk

- XML to describe "job-ticket" information
  {JDF (Job Definition Format)
  PJTF (Portable Job Ticket Format), et al.} (PODI's wrapper standard)
- PPML (Personalized Print Markup Language)
- Ebooks (except as another useful output format)
- ECommerce, eBusiness, B2B, B2C, new business models
- Syndication of content (PRISM et al,)
- Physical interchange of XML and packaging
  (SOAP, XML-RPC, etc.)

slide 3

Where We Are Going Today
XML as Content

XML as the text/data/content that is being printed

- What is XML and how it works
- The purpose of XML: Multipurpose
- Production of printed material from XML
- Inserting XML into the publishing cycle
  - How XML Works
  - Where XML may fit into the publishing cycle
- Other print issues and XML
- XML Information Resources

slide 4

A Quick Poll (Who You Are)

How many of you are (or manage people who are)...

- Compositors or typesetters
- Designers (whatever that means)
  - Print publications
  - Web publications
- Publishers of
  - Books (monographs, reference series, etc.)
  - Journals
  - Technical Documentation
  - Catalogues
- Trainers or training publishers (CBT, web, textbook, etc.)
- Content providers

slide 5

A Quick Poll (What You Know)

How many of you know:

- HTML
- XML or SGML
- PDF
- Quark
- Microsoft Word Templates
- Other high-powered composition systems such as Miles 33, Penta, or 3B2

What Is XML?

slide 6

The Word "XML" Is Used to Mean:

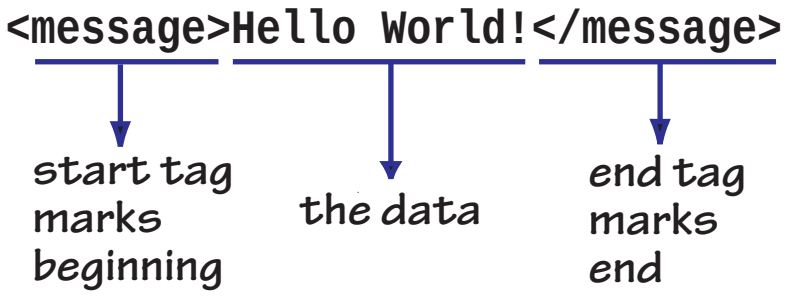
- An open standard (well ... a W3C recommendation) that provides:
  - A data format
  - A data modeling language
- The use of XML-formatted data in an application (like a browser)
- A metalanguage for creating markup languages
- A set of associated recommendations and specifications
  (link, style, transformation, query, APIs, etc.)

slide 7

XML Works through Tags

Paired tags:

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



slide 8

XML Documents

- In XML jargon, your data (no matter what form) is called a "document"
- A document is a coherent, ordered collection of information:
  - reference book
  - journal article
  - trade paperback
  - cover or dust jacket/flap copy
  - sales catalog
  - database load file
  - invoice

slide 9

How XML Looks At Data

Documents

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

slide 10

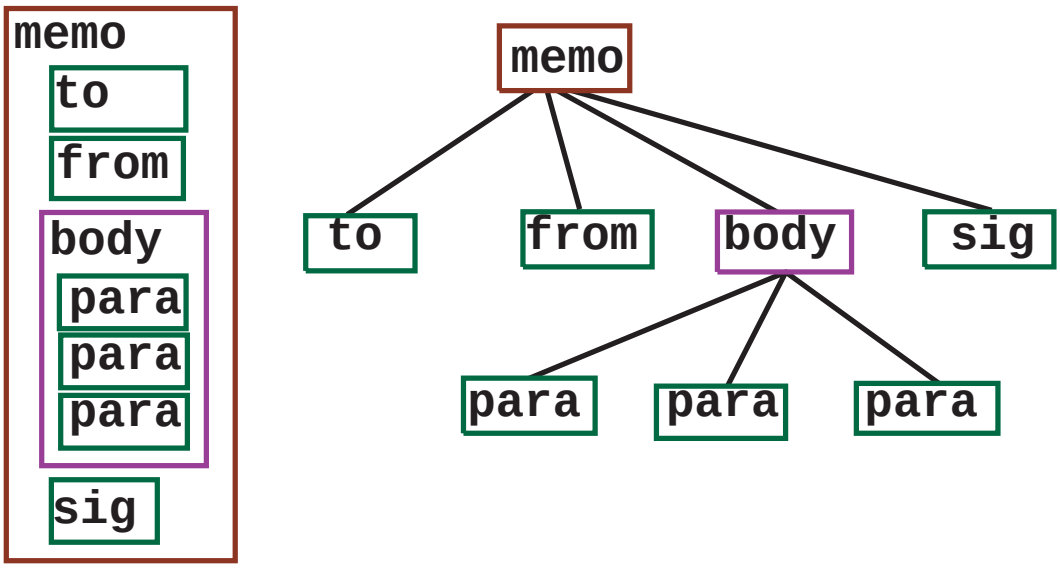
XML Elements

An *element* is an identifiable, named component of a document (payment term, paragraph, part number, title, author, unit price, bulleted list)

- Can have content (data, other elements)
- Can be a pointer to information (hypertext link, table reference)
- Must be contiguous (one start and one end; no holes in the middle)

slide 11

Elements Can Nest



slide 12

Attributes Add Further Description

- Live inside start tags
- Say something *about* the data
- Add information to the our knowledge of the element

```
<phoneNumber type="unlisted" rate="premiumplus"
   assigned="1996-04-01">301/315-9631</phoneNumber>
```

slide 13

XML Isn't:

- A programming language
  Does not replace C++, Java, perl, Python ...
- A user interface
- A standard set of tags
- A recommended set of tags
- *A presentation format*
- *A formatting or processing system*

The Vendor-specific File Trap

slide 14

The XML Cliche

There is always a

- Faster
- Cheaper
- Easier
- More exact

way to do any ***one*** thing than using XML.

slide 15

But, If You Create Content in a Blackhole Application...

Consider the output of these applications

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

These applications typically produce

- Their output format (for example, postscript)
- Their internal file format (for example, `.doc` for Word)
- A small number of other formats (ASCII)

slide 16

What Data Formats Do the Following Produce?

- QuarkXPress
- Photoshop/Illustrator
- GoLive
- Indesign
- PageMaker
- Microsoft Word

slide 17

Vendor-specific Formats Belong to the Vendor

- Embed “codes” in the data to drive format
- Rely on specific software to work with that format
- Such codes:
  - *Can be changed by the vendor at any time.*
  - Can't be changed by you, unless you are willing to maintain the additions, in the face of updates by the real vendor
  - May make it difficult or impossible to
    - Extract data in usable form
    - Make global style changes
    - Rearrange/extract/reuse/repurpose data

slide 18

The New Cliches

Repurposing and Reuse Reigns

- Print is not enough any more
- Single-use data is too expensive
- Information is a corporate resource and must be managed accordingly
- Pages are not enough anymore; every user needs a personalized look and feel
- Web design and print design are different

slide 19

Why Companies Are Switching to XML

- Version control and configuration management
  (Manage text and data revisions as software is managed)
- The document/content repository as a corporate resource
- Electronic slice and dice for on-demand publishing
- Ability to store metadata with the data
  - Who wrote it?
  - When was the last update?
  - Who checked it?
  - What's the source?
  - has legal approved it for publication?
  - What are the print and electronic permissions?

slide 20

Ultimate Purpose of XML

- Encode (mark up) data only once
- Produce many products from that markup
- Enable semantically complex searching
- Reuse data (in whole or part) many times
- Interchange data freely
- Enable machine-to-machine communication
- Let whole communities agree on data content
- Let data live a long time

slide 21

Use XML When the Same Content Must Be

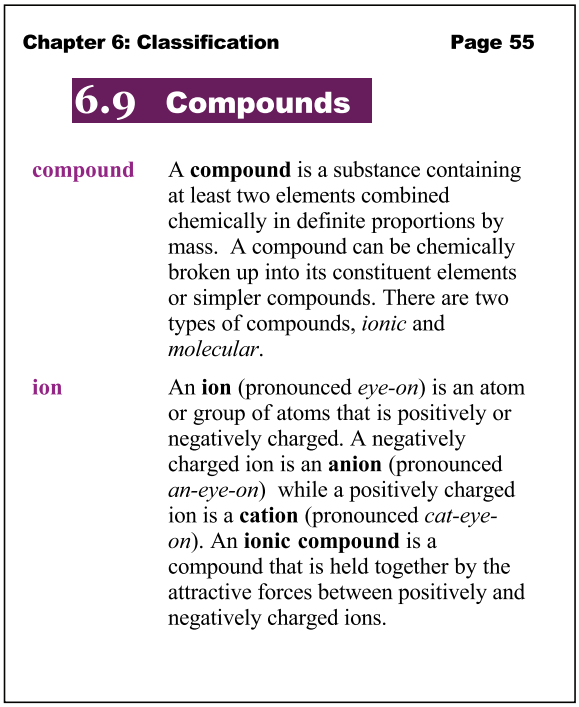
- Printed for our {book, journal, magazine, etc.} as we always have
- Published on our website
- Sold to content aggregators
  (such as meansbusiness.com, books24x7.com, EBSCO, Mead)
- Put on our intranet for electronic review and revision
- Put out on the CD-ROM set
- Placed in the data repository, so we can reuse, repurpose, check rights and permissions

One XML **"Text Book"** and Many Results

```
<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>
...</section>
```

We Still Print Textbooks



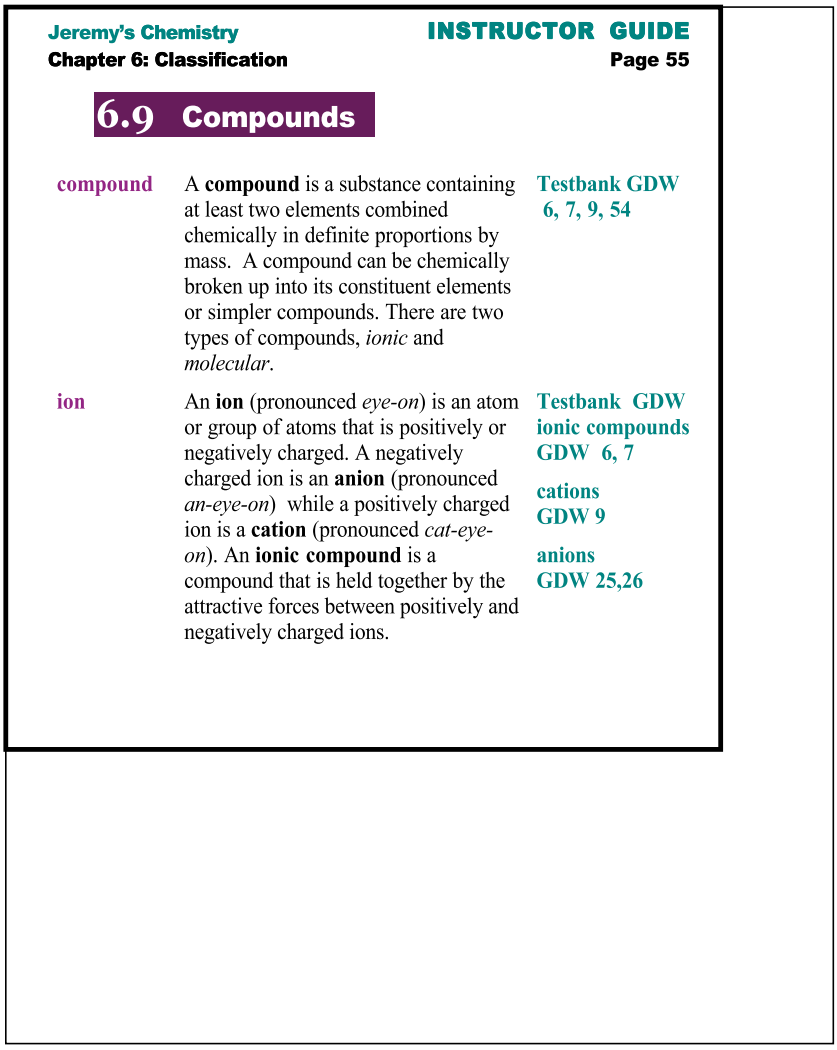
Chapter 6: Classification Page 55

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

Textbooks May Have Instructor's Manuals



Jeremy's Chemistry
Chapter 6: Classification

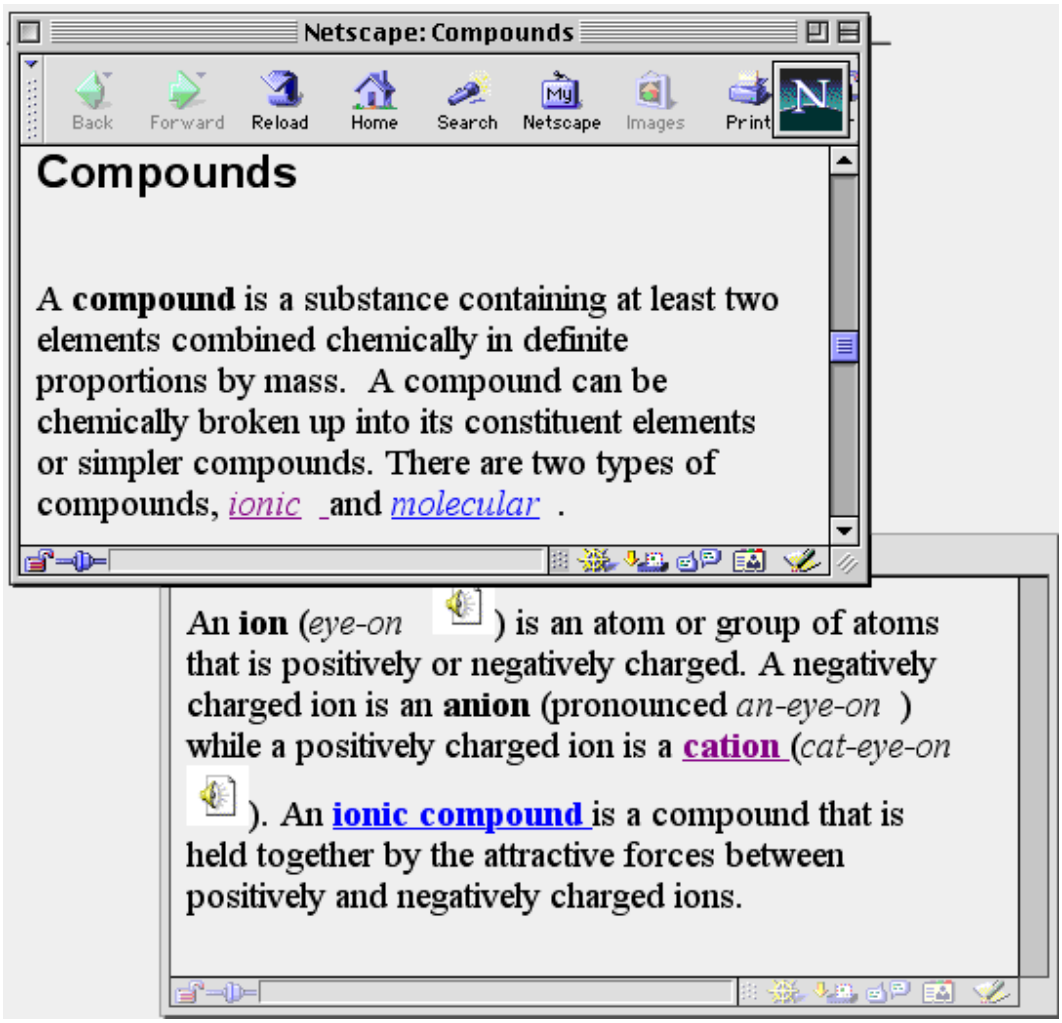
INSTRUCTOR GUIDE
Page 55

6.9 Compounds

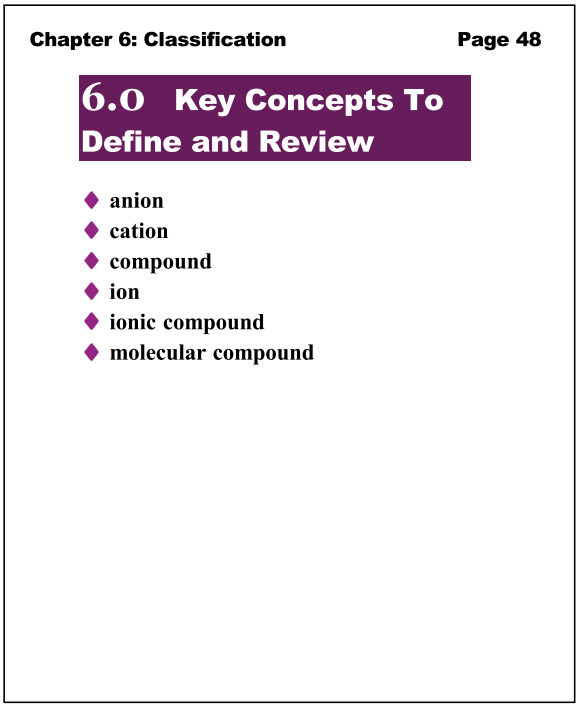
compound	A **compound** is a substance containing at least two elements combined chemically in definite proportions by mass. A compound can be chemically broken up into its constituent elements or simpler compounds. There are two types of compounds, *ionic* and *molecular*.	Testbank GDW 6, 7, 9, 54
ion	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 ion is a **cation** (pronounced *cat-eye-on*). An **ionic compound** is a compound that is held together by the attractive forces between positively and negatively charged ions.	Testbank GDW ionic compounds GDW 6, 7 cations GDW 9 anions GDW 25,26

View This in a Web Browser/eBook

Convert into HTML or HTML-like format (today) XML tomorrow



Automatically Generated Section of Same Textbook



Chapter 6: Classification Page 48

6.0 Key Concepts To Define and Review

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

slide 27

Same Source, Different Results

- On the Web, eBook, and CD versions:
  - Tie the pronunciations to audio files
  - Link keywords to definitions in the dictionary
- Make large print, voice synthesis, and Braille
- Collect statistics on which test questions are used, how often, and where

slide 28

Reuse for Display

- Print
- Voice synthesis
- Braille
- Electronic — including HTML

Slice & Dice Publishing

A Dream, possibly realized with help from XML

Slice & Dice reuse is:

- A hot topic
  - Publishers dream of combining related resources to make new publications
  - Assume XML will make this possible
- Technologically achievable
  - XML allows combination of info from many sources
  - Formatting controlled for this use or display
  - ToC and Index generation from XML automatically
- Editorially challenging
  - Books have structure
    - Topics build on one another
    - Later sections assume content of earlier
  - Voice, point of view, level of detail vary from source to source
  - Few materials written as stand alone. If they were, combining them makes choppy book.

From XML to Typography

How Print Is Produced from XML

Formatting: Remember What XML Looks Like!

```
<?xml version="1.0"?>
<RESUME>
<CONTACT.INFO>
<NAME>Heinrich Rudolf Hertz</NAME>
<ADDRESS>Bonn, Germany</ADDRESS>
</CONTACT.INFO>
<OBJECTIVE>To continue researching electrical discharges in
rarefied gases in an academic setting.</OBJECTIVE>
<SUMMARY><PARAGRAPH>Over ten years academic research
studying electromagnetic waves.</PARAGRAPH>
</SUMMARY>
<WORK.EXPERIENCE>
<JOB.BLOCK>
<EMPLOYER><NAME>University of Bonn</NAME>
<LOCATION>Bonn, Germany</LOCATION></EMPLOYER>
<DATES>
<START.DATE>1889</START.DATE>
<END.DATE>1894</END.DATE></DATES>
<JOB.TITLE>Professor of Physics</JOB.TITLE>
<RESPONSIBILITY>Research the discharge of electricity
in rarefied gases</RESPONSIBILITY></JOB.BLOCK>
<JOB.BLOCK>
<EMPLOYER><NAME>Karlsruhe Polytechnic</NAME>
<LOCATION>Germany</LOCATION></EMPLOYER>
<DATES><START.DATE>1885</START.DATE>
<END.DATE>1889</END.DATE></DATES>
<JOB.TITLE>Professor of Physics</JOB.TITLE>
<ACTIVITY>Produced and studied electromagnetic waves
(radio waves), confirming Maxwell&apos;s electromagnetic theory
</ACTIVITY>
<ACCOMPLISHMENT>Established that light
and heat are electromagnetic waves (1887); first to produce
radio waves artificially.</ACCOMPLISHMENT></JOB.BLOCK>
</WORK.EXPERIENCE>
<EDUCATION>
<SCHOOL><NAME>University of Berlin</NAME>
<DEGREE>Ph.D. magna cum laude</DEGREE>
<PROGRAM>Physics</PROGRAM>
<GRANT.DATE>1880</GRANT.DATE></SCHOOL>
</EDUCATION>
<PUBLICATIONS><PARAGRAPH><TITLE>Electric Waves
</TITLE> (1893);<TITLE>Miscellaneous Papers</TITLE> (1896);
<TITLE>Principles of Mechanics</TITLE> (1899)</PARAGRAPH>
</PUBLICATIONS></RESUME>
```

slide 31

XML Separates Content from Format/Behavior

How it looks (16 pt Helvetica Bold) or what it does (starts a javascript)

- Is based on the tagging
- Is the same for every tag *in the same context*
  - NOT one tag per one format
  - Title in the table may differ from Title in the figure or title in the chapter

slide 32

What We Might Like to See in Print

Heinrich Rudolf Hertz
Bonn, Germany

Objective: To continue researching electrical discharges in rarefied gasses in an academic setting.

Summary: Over ten years academic research studying electromagnetic waves.

Experience:

1889—1894	University of Bonn, Bonn, Germany Professor of Physics Research the discharge of electricity in rarefied gasses
1885—1889	Karlsruhe Polytechnic Professor of Physics Produced and studied electromagnetic waves (radio waves), confirming Maxwell's electromagnetic theory. Established that light and heat are electromagnetic waves (1887); first to produce radio waves artificially.

Education:

slide 33

What Has to Happen to Get from Tags to Pages

- Formatting flows from tags
- If it should look different, it needs a different (choose one)
  - Tag
  - Attribute
  - Context
- Hand adjustment still needed
  (page or column balance, graphics placement, aesthetics, etc.)

slide 34

How To Get from Tags To Pages

- Hand it to a service (let a compositor handle it)
- Convert the XML to instructions for a composition system (like Quark, PageMaker)
- Use a “native” XML or SGML composition system (like Penta, Miles 33, 3B2)
- Use a “native” XML or SGML desktop publishing system (like FrameMaker + SGML)
- Use XSLFO (XSL Formatting Objects)

slide 35

Most Options Need an Output Specification

(Frequently called “Stylesheet”)

- Says what XML data will look like or how to behave
  - On screen or paper
  - Or in other media (for example in audible output)
- Defines an appearance or rendition or behavior
  - For each element
  - In each of its contexts within a document

slide 36

Stylesheets are the Bane and the Golden Opportunity of XML

- The bane is you have to have one
- The opportunity is you may have many!

slide 37

The Opportunity Works Both Ways

- One stylesheet, many documents
  - Maintains consistency of format (“look and feel”) across documents
  - Is easy to develop, maintain, and apply (house style)
- One document, many stylesheets
  - Allows for different media types: print, on-line, etc.
  - Is easy to produce derivative documents: selections, summaries, indexes, catalogs ...

Where XML Can Be Used in Publishing

Parts of an XML Application

slide 38

Logical Components of an XML Application

- XML document (tags and text)
- DTD or Schema (the model)
- Output Specifications (how looks/behaves)
- Transformations (from here to there)

slide 39

Component: XML Document

The tags (markup) and the text (content)

- Two types
  - Well-formed (syntactically correct)
  - Valid (is syntactically correct and matches a model)

slide 40

Component: The Document Model— DTD (Document Type Definition) or Schema

- A model for one type/class of information (a “document”) (reference book, bank transfer, journal article, memo, help-topic)
- Is a set of rules describing how documents of that type can be marked up
- Is written in the formal syntax of XML

slide 41

DTDs/Schemas Express Rules

for example:

- **Reference book** =
  Book-level Metadata followed by
  Front Matter followed by
  Body followed by
  optional *Back Matter*
- **Purchase Order** =
  Order Header followed by *List of Order Detail*, followed by
  optional *Order Summary*
- **paragraph** = data characters and may include any of the following: *Person Name*s *URL*s, and/or *Geographic Region*s

slide 42

Why Use a DTD or Schema?

- DTD is a contract between producers and consumers
  (Both can validate to see if they got/sent what they expected)
- Formal specification of information *types* allows consistent downstream processing
- Supports interoperable families of documents
  - Ensure that information conforms to model (validation)
  - Parties don't have to share software or applications

slide 43

To Share Information, Share a DTD

- Publisher communicates to conversion house
- Content provider explains tagging to
  - Compositor for typesetting
  - Web designer for building website
  - Database or repository designer
  - Software vendor for customization

slide 44

How to Tell a DTD from a Schema

DTDs are the model the W3C XML Spec describes

- Syntax like SGML DTDs (unique and obscure)
- All current XML tools use these

DTDs look like this

```
<?xml version="1.0"?>
<!DOCTYPE memo [
<!ELEMENT memo (to, from?, body) >
<!ELEMENT to   (#PCDATA)        >
<!ELEMENT from (#PCDATA)        >
<!ELEMENT body (#PCDATA)        >
]>
```

slide 45

How to Tell a Schema from a DTD

Schema are proposals for DTD replacement/enhancement

- Schemas promise
  - To use the element/attribute syntax of documents
  - Provide all the functions of a DTD
  - Also provide many things including:
    - Strong data typing (date, time, integer, string, etc.)
    - Inheritance mechanisms
    - Default and required values for content

There are many different schema languages

- XML Schema (W3C, not finished)
- XML Data (Microsoft XDR)
- Relax, TRex, Schematron, SOX, DCD, etc.

slide 46

Schemas Look Like This

```
<xsd:element name="memo">
    <xsd:element name="to"></xsd:element>
    <xsd:element name="from" type="xsd:string"
          min0ccurs="0" maxoccurs="unbounded"/>
    <xsd:element name="body" type="xsd:string"/>
</xsd:element>
```

-

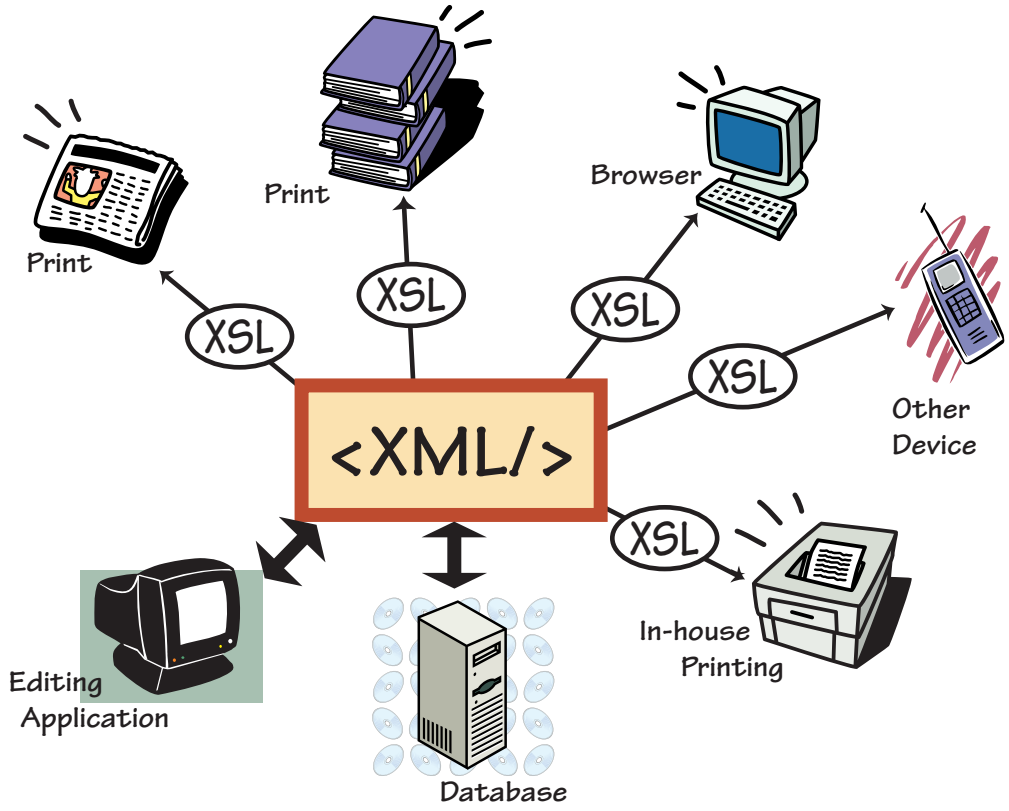
Component: Output Specification Stylesheets

- Contain styles for XML document display and use
- At least one is necessary
- Opportunity: you may have many different stylesheets
  - Print
  - Large-type print
  - Internal (editing) website
  - Website
  - Aggregator or search service

Component: XML Transforms

slide 48

XML and XSL



slide 49

XSL For XML Transformation (XSLT)

Transforms from one set of tags directly into another
Transform XML into

- HTML for browsers
- Other (XML) tag sets for further processing
- Plain text formats (e.g., loader files for databases)
- Non-XML tag sets

Where Does XML Fit in the Publishing Process?

slide 50

The Myths

- "You Can't Create Quality Typography from XML"
- XML documents look like they came out of a typewriter
- You have to give up H&J with XML

- *XML does not provide sufficient control for*
  - *Kerning pairs*
  - *Color separation*
  - *Complete WYSIWYG proofing*

slide 51

The Truth: What XML Does

- Formatting is based on tags
- *This* tag in *this* context has this format
- If something needs to look different, there must be a distinction in the markup
  - A different tag (`<bold>` versus `<italic>`)
  - An attribute (`<list style="numbered">`)
  - A different context (`<title>` inside table, or figure)
- No markup distinction; no formatting distinction

slide 52

The Truth: What XML Should Not Do

- Page layout and text flow into objects
- Hyphenation and Justification
- Column balancing
- Floats and keeps (but metadata may advise)
- Windows and orphans (but metadata may advise)
- Real color (in all its meanings)
  although tagging may suggest (like `color="BK#0000"` in HTML)
- Illuminated initial caps
- Selective spreading of vertical justification

slide 53

The Reverse Myths

- XML doesn't need skilled typographers; it has style sheets
- Formatting based strictly on tags is good enough;
  we can do*lights-out* composition
  (This last is usually said by composition vendors.)

slide 54

The Truth

- You *can* make type from XML look just like a typewriter produced it
- When skilled typographers design style sheets,
  the product looks better
- Lights-out
  - Can work with large amounts of repetitious data
    (catalog publishing, directories, phone books)
  - Does *not* work for real typographic quality

slide 55

XML in a Possible Print Production Life Cycle

Bold indicates where XML might be used

1. Business case is made
2. **Product structure design**
3. Content is solicited/Contracts/etc.
4. **Manuscript content created**
5. Presentation *design(s)* created
6. **Content is edited for truth, beauty, style (proof cycles with author)**
7. **"Final" content copyediting and style corrections**
8. **Pages produced**
9. **Still another proof cycle**
10. **Final corrections**
    ----
11. Final page proofs produced,
    tweaking by hand to“fix”typography or create work of art
12. Several magical steps of pre-flight and color work and etc.
13. Print run produces book
14. PDF is made for the electronic book
15. HTML is made for the web (from XML)

slide 56

XML Feeds the Print Production Cycle

XML does not participate in the pagination and fine typography process

- "By the Time I reach the Prepress Stage..." XML is GONE

XML has nothing to do with

- Particular output formatting or behavior such as
  - *Kerning*
  - *Color*
  - *Column Balancing*

slide 57

Two Main Ways XML Typography Can Work

XML feeds proprietary formatters

- XML input is mapped to internal format
  (XML translated into Quark, XyVision, FrameMaker, etc.)
- Proprietary codes accomplish typography

XML-based composition engines

- System works natively in XML
- Proprietary codes hidden in XML processing instructions

XML in the Input Data Side

slide 58

Who Tags the Data

- Authors (in-house or external)
- Copy-editors
- Production staff
  - Possibly special “taggers”
  - Production Editors or other Production staff
  - Compositor's production people
- Conversion vendor
- One or more computer programs

slide 59

When Is the Data Tagged

- During creation/authoring
- Between author and any edit
- As part of editing
- As part of production
- Post Production
  - As extension to process (for example by compositor before return to publisher)
  - When placed into repository
  - Long after production, possibly by years

slide 60

The Best Place in the Cycle to Use XML

Greatest benefits mean tagging as early as possible in the production cycle

- Pre-repository brings searching benefits
- Pre-production brings electronic reuse and control (XML inside content management repository)
- Pre-editing allows hyperlink editing/checking, snazzy electronic edit features, electronic communication with authors, etc.

slide 61

Return of the Galley?

- XML is best used early in production
- Pages flows from tags and are content-proofed
- Final format proofing (column balancing, etc.) is postponed as long as possible
  - So that all changes are in XML source
  - Because format changes may involve the non-XML portions of the system

slide 62

XML in Checking and Proofing Cycles

- Cannot really edit XML in WYSIWYG form
- Edit XML in WYSIROTS
  (What You See Is Representative Of The Structure)
- Line between content edit and copy edit blurs
  (bad typography usually means bad tagging or missed tagging and these may be content tags)

slide 63

Different Checking and Proofing

- Level changes
  - Less worry about transposed letter, missing punctuation
  - More worry about missing structures
- Specific checks change
  (don't look for comma, look for `<author>`)
- One error may show many times but still be just *one* error
  - There are two commas after every date in every bibliographic reference
  - All second level heads are italicized
- Check the same element in different contexts

slide 64

New Checking and Proofing Possibilities

XML allows for more intelligent proofing than just "Does it look right?"

- Lists of any element and many combinations
- Print content of reference next to place where reference is made
- Some automated link checking
- False color proofs
  - Add color to make things stand out
  - Add numbering
  - Add links to what needs to be checked

Publishing XML Data

slide 65

Publishing Relational Data

- Data stored in ordinary relational database
- Extract data using SQL queries or reports
- Wrap tags around data as extract producing XML
- Transform or format XML source to produce
  - PDF
  - Postscript
  - Proprietary word-processor format such as Quark, .RTF (MS Word), or .MIF (FrameMaker)
- Example: catalog or directory data

slide 66

Content Management Using XML Repository

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

slide 67

XML Content Repository as a Resource

- Content management at many levels of granularity
- Data-base-like check-in and check-out
- Real version control
- Integrate graphics management systems and content management systems

Some Differences Between XML and Other Markup

slide 68

XML Is Not Like HTML or Proprietary Markup

Unlike HTML or proprietary typesetting languages

- Number of tag types is infinite
  (Structure tags not limited to h1, h2, p, etc.)
- Tags may describe content as well as structure
- Tag combinations may be constrained by DTD, schema
- Tags
  - Usually do not indicate format/behavior
  - May *not* relate to format/behavior

slide 69

These Differences Can be Categorized

- Types of Markup
- Consequence of Structure
- Consequences of DTDs/Schemas
- Consequences of Stylesheets
- Formatting consequences of markup

slide 70

Types of XML Markup

- Content of the data
- Structure of the document
- Value-added information
  - Location and Navigation
  - Metadata
- Rendering/Processing information (presentation and formatting)

slide 71

Content Markup

What type of information is this?

- Environmental Impact
- City, state, zip code
- Question, answer
- Methodology section
- Part number
- Executive Summary
- Drug Dosage section

slide 72

Structure Markup

What part of the document is this?

- Paragraph
- Title
- Figure
- Chapter
- Table
- Signature block
- List
- Bibliographic header

slide 73

“Location or Navigation” Information

Added to text to make it more functional, useful or manageable

- Hypertext links
- Cross-references
- Indexing terms

slide 74

Metadata (Data About the Data)

- Bibliographic information
- Index or search terms
- Revision or version
- Status and workflow tracking information
- Data source
- Editor's or reviewer's comments
- Abstracts, teasers, cataloging data

slide 75

Rendering/Processing Markup

How text should behave, display, or print

- A complete script to call and perform
- Iconization
- Position of graphics on the page
- Line breaks in titles
- Visual or auditory highlighting
  (sometimes a word is **bold** just because the author said so)

slide 76

More of Certain Kinds of Tags

- More wrapper containers
  - Typography drivers versus
  - Electronic cut and paste and reuse
- All elements are containers rather than flat structure

Elements as Containers Rather than as a Flat Sequence

slide 77

HTML Structure Is Flat

- *Title*, followed by
- *Paragraph*, followed by
- *Heading Level 1*, followed by
- *Paragraph*, followed by
- *Paragraph*

(Most word processors and desk-top publishing applications are like this, too)

slide 78

XML Makes Nested Structures

- Tags identify the start and end of each structure, *not*simply the start of a format
- A document might contain:
  - *Title* followed by
  - *Paragraph* followed by
  - *Section*, containing
    - *Title* followed by
    - *Paragraph* followed by
    - *Paragraph* followed by ...

Consequences of DTDs/Schemas

slide 79

New Error Category: Model Validity

Document may need to match document model

- Solutions include
  - Change the data to match the DTD
  - Change the DTD to match the data
  - Have a loose DTD for production emergencies
  - Use “well-formed” not valid documents

slide 80

New Error Category: Looks Fine; Isn't Right

- Incorrect non-display data (metadata, index terms)
- Missing non-display data (identify all drug names, all people)
- Tag Abuse
  - Choose tag to make it look right
  - Ignore definition of element and proper use

Consequences of Stylesheets

slide 81

Two Potential Sources of Error

- Source document error (more common)
- Stylesheet Error (larger errors, easier fix)

slide 82

Source Document Error

- Usually only one or two
- Not a consistent pattern
  - This 4th level head looks like a 2nd level
  - This word should be italics
  - Half of this sentence is missing.
- Solutions
  - Source data is wrong
    (Fix this one tagged item)
  - No way to make tagged item “look right”
    (Style sheet may need fixing)
  - Tagset is inadequate (oops)

slide 83

Stylesheet Error

- Error flows from correct tagging
- Consistent pattern
  - All 4th level headings look like 2nd level.
  - The first item in each bulleted list is missing.
  - All genus and species names are in bold not italic.
- Solutions
  - Tags are fine; don't change source
  - Figure out the pattern
  - Get the stylesheet fixed

Formatting Consequences of XML Markup

slide 84

Markup Is Not Equivalent to Format (Part 1)

Different Markup, But the Same Format

- All these are “section-level” elements and print the same way
  - `<environmental_impact>`
  - `<methodology>`
  - `<exec-summary>`
  - `<drug-dosage>`
- All of these print as italic, small caps
  - `<bibliographic-reference>`
  - `<drug-name>`
  - `<corporate-division>`

slide 85

Markup Is Not Equivalent to Format (Part 2)

Same Markup, Different Format *by Context*

- Each type of `<title>` is distinct
  - Title of a table
  - Title of a chapter
  - Title of a figure
  - Title of the book
  - Person's hereditary title

slide 86

Vanishing Markup: Markup Has No Formatting Consequences

- The content of these tags *appears* exactly like the other, untagged words in the same paragraph
  - `<person-name>`
  - `<geog-region>`
  - `<drug-name>`
  - `<tradename>`
  - `<keyword>`

slide 87

Vanishing Data: Marked Up Material Does Not Print

- Marked up text (inline with other text) that does not print or display at all
  - Index or search term embedded inside text
  - Journal header information for a journal article
  - Version control or workflow information
  - Metadata
    - Embedded within text for searching
    - At the front of an element for identification
    - Associated with an element to provide additional control information

slide 88

The Data/ Document Distinction Is Blurring

- "Documents" generated from databases
- Data resides within the text of the document
- Data/document combinations becoming common
  - Oxford University Press American National Biography
    - Half data
    - Half textual article

slide 89

Format may Flow from Containment or Position

- One XML view of list markup

```
<paragraph>
   aaaaa aaaa words in a paragraph aaaa aaaa aaaa
   aaaaa aaaa words in a paragraph aaaa aaaa aaaa
   aaaaa aaaa words in a paragraph aaaa aaaa aaaa
      <BulletedList>
      <item>first bulleted list item
      <item>subsequent bulleted list item
      <item>subsequent bulleted list item
      <item>subsequent bulleted list item
```

- Format may be associated with the whole list, or
- Format may be associated with the first child of the list, last child of the list, all the children

Aside: Spontaneous Generation of Data: Typically Called “Generated Text”

slide 90

Generated Text

Text that is not in the data, but is put in by the display or formatting system, based on the tagging

For example:

- The numbers in a numbered list (1., 2., 3.)
- The bullets in a bulleted list
- The enumerator on a footnote
- Chapter 1.
- Figure 3.6
- (See Figure 3.4: All Cars Eat Gas)

Authoring Systems *Should* Generate Text

(XML Makes This Easy)

- Consistency is assured
  - For each document
  - Over time for many documents
- Changes:
  - Automatically appear to all documents
  - Made once appear everywhere
- Author:
  - Can’t make it wrong (or better!)
  - Has less work to do
  - Can use GUI while system fills in gory details (cross-references)

Should Archival Systems Use Generated Text?

- Is the stylesheet necessary to know what was actually produced?
- Can you search on “generated” words?
- If the text is not all there, can the document legally be copy-of-record?
- Is there a requirement to reproduce the document exactly as it was on a particular past date, thus requiring knowledge of all stylesheets and the system on that date?
- Is vital information hidden in the generated part?

```
TAGGED TEXT                    DISPLAY
   <expn>aaaa</expn>                BAD EXAMPLE  - "aaaa"
   <expg>bbbb</expg>                GOOD EXAMPLE - "bbb"


   <exp time=b">ggg</exp>           BEFORE: ggg
   <exp>fff</exp>                   AFTER:  fff
```

slide 93

Archive Comparison

- Use PDF (or similar) when:
  - There is one piece
  - Unit is the page, reuse on that level
  - Look and feel is as author intended
- If archive in XML, then:
  - There may be many parts as well as an external stylesheet
  - Some text may be generated, thus not there for search or raw comparison
  - Look and feel is
    - as stylesheet writer(s) intended
    - lost

Other Issues With XML and Print

slide 94

All XML Is Not Created Equal

- XML tag design may be optimized for
  - Print
  - The Web
  - Customized search program on CD-Rom
- More complicated tagging has more content tags
  - The more work and $$ to tag
  - The more work to create print

slide 95

Make Print from Structural Tagging

```
Book
   Title
   Chapter      (may repeat)
      Title
      Section   (may repeat)
         Title
         Para   (may repeat)
         Section (may repeat)
   Appendix     (may repeat)
```

slide 96

Make Print from Content Tagging

```
Book
   Title
   Executive Summary
   Introduction
     Preface
     Background
   Methodology
   Results
   Conclusion
   Appendix     (may repeat)

Each of the section types
      Title
      Section   (may repeat)
         Title
         Para   (may repeat)
         Section (may repeat)
```

slide 97

Converting Your Print Files to XML

- What kind of XML?
  - Structure-only (heads and text) may be automatable
  - Content tagging may require subject experts
- Can heavy subject tagging be added post-production?
- Mismatch between typographic requirements and tagging granularity

slide 98

When Making Electronic Files from Print

(Or Making Both from the Same Source)

- Markup
  - May be richer
  - May have many tags in the place of one
  - May use many “extra” tags

slide 99

"Extra" Tags May Include

- Index terms embedded in the text, not gathered at the end
- Container elements
- Multiple graphics
  - The same graphic in both black and white for print and color for electronic
  - The same graphic at various resolutions
  - Additional still graphics for electronic only
  - Animations, videos, etc. for electronic only

slide 100

Some Cost Implications

- If you create a product for print first and make electronic products from it later
  - Total costs go up
  - Immediate costs go down
  - Electronic material (such as hyperlinks) must still be checked and proofed
  - There is very little processing benefit

Remember: a file that typesets cleanly **may not** be a clean electronic file

The Reverse Cost Implications

- If you optimize for electronic now and create print simultaneously
  - Total costs go down
  - Immediate costs go up
  - Quality may increase a lot
  - Electronic additions may add a lot of time

Remember: the benefits are long term

- Increased opportunity
- Building corporate resources
- Ability to create new products more easily

The expense and hard work are immediate

slide 102

Costs and Benefits Not Equitable

(Warning: Costs in budget of Department A, benefits accrue to Departments C and D or to the company as a whole)

- Some groups will have increased workload, other groups will find production faster and easier
- Expense and hard work are immediate, specific to a group
- Most benefits
  - Increase the bottom line for a different group
  - Are long term
  - Are corporate-wide or sales related
  - Are cumulative (your job takes 2 months longer, but our jobs are: done in half the time, much cheaper, or only possible now)

XML and "Special" Characters

slide 103

"Special" Characters

Special characters include:

- Other alphabets such as Greek, Cyrillic
- Special symbols such as paragraph and section symbols
- Smiley faces, clubs, hearts, and other "dingbats"
- Mathematical and chemical symbols
- Ordinary accents and diacritical marks such as acutes, graves, umlauts

slide 104

XML and Characters

- XML is (by definition) based on a defined set of characters called "Unicode"
  - Most known alphabets
  - Some special math and scientific characters
  - Some publisher's symbols
- A character in an XML file may be represented as:
  - A named thing (called entity references)
    `&trade;` is a trademark symbol,
    `&copyr;` is the copyright symbol
  - A numbered thing (Unicode character references)
    `&#x2122;` is a trademark symbol,
    `&#169;` is the copyright symbol

slide 105

How to Deal with XML's Characters

- Ideally, operate in Unicode
- Make translations from the entity references to your "codepage", "fontset", whatever
- Make similar translations from the Unicode character entities *that you use*
  (Almost nobody needs all of Unicode)
- If you must — punt— a character can be a graphic

A Few Representative XML Composition Systems

slide 106

Representative Composition Engines

The following are among the companies making high-end tools that can compose type from XML (alpha order)

- 3B2 (Advent & Lightbinders)
- DL Pager/DL Composer (DataLogics)
- Oasys and Genera (Miles33)
- SGML Publisher (Penta — Version 3.0 and up do XML too)
- XML Professional Publisher (XyEnterprise)

slide 107

Representative Desktop Engines

The following are among the desktop tools that can compose pages from XML (alpha order)

- BladeRunner (Interleaf)
- DeskTopPro (Penta)
- Epic + Publisher (Arbortext)
- FrameMaker + SGML (Adobe)
- PowerPublisher with UltraXML (WebX Systems; supports XSLFO)
- Quark (needs avenue.quark and/or Roustabout)

slide 108

Other XML Print Engines

- Word-Processor based
  - WordPerfect (Corel)
  - Microsoft Word plus add-ons (WORX, etc.)
- NuDoc (BitStream)
- TopLeaf XML Composition System (Metaformix Information Systems & TurnKey Systems — promises looseleaf)
- inForm Xprint and Xrender(Prout AG)

slide 109

The Question in XML Publishing Packages Is Round-Tripping

- Round-tripping means
  - Taking in XML as input
  - Producing pages
  - Producing clean XML as output that includes any changes introduced during production)
- Can the packages listed do that?
  - Yes, they can
  - Some by reverse translation from their own format
  - Some by using XML as their internal format

slide 110

Can All of Them Really Do Round-Tripping?

Well...

- Some of them require "clean, well-structured input"
- Some (such as Quark and WordPerfect) require that you use styles
- Some of them can remove their own processing instructions and some cannot

All of them require setup!

slide 111

Look at the High End List One More Time

Weren't these companies in the typography game way before XML? (and even before SGML in 1986?)

- Penta
- 3B2
- Miles33
- DataLogics
- XyEnterprises

If they created real typography then, why would XML source change that?

Where to Get More Information

slide 112

The Source for XML and Related Information

- Robin Cover's SGML/XML Web Page: **`http://www.coverpages.org`**

slide 113

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)
- http://www.xml.org

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.

slide 115

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, Extreme Markup Languages, XML 2001)
- XML.COM: **`http://www.xml.com`**

slide 116

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`**

Appendix 1: Acronyms Used in Ths Talk

API	**Application Program Interface**
B2B	**Business-to-Business**
B2C	**Business-to-Customer**
CML	**Chemistry Markup Langauge**
CSS	**Cascading Style Sheets**
DOI	**Digital Object Identifier**
DOM	**Document Object Model**
DSSSL	**Document Style and Semantics Specification Language**
DTD	**Document Type Definition**
eBook	**Electronic Book**
ebXML	**Electronic Business XML**
EDI	**Electronic Data Interchange**
GUI	**Graphical User Interface**
HL7	**Health Level Seven Initiative**
HTML	**Hypertext Matkup Language**
ISBN	**International Standard Book Number**
MathML	**Mathematics Markup Langauge**
OeB	**Open eBook Specification**
OO	**Object Oriented (applied to database)**
PDA	**Personal Digital Assistant**
PDF	**Portable Document Format**
PODI	**Print on Demand Initiative**
PPML	**Personalized Print Markup Langauge**
PRISM	**Publishing Requirements for Industry Standard Metadata**
RPC	**Remote Procedure Call (e.g., XML-RPC, SOAP, etc.)**
RTF	**Rich Text Format**
Sax	**Simple Application Profile for XML**
SGML	**Standard Generalized Markup Langaage**
SOAP	**Simple Object Access Protocol (Microsoft)**
SQL	**SEQUEL Query Language**
SVG	**Scalable Vector Graphics**
UML	**Universal (Uniform) Modeling Language**
URL	**Uniform (Universal) Resource Indicator**
W3C	**World Wide Web Consortium**
WWW	**World Wide Web**
WYSIROTS	**What You See Is Representative Of The Structure**
XML	**Extensible Markup Language**
XSL	**Extensible Stylesheet Language**
XSLFO	**XSL Formatting Objects**
XSLT	**XSL Transformations**