

# Electronic Commerce Connection

## **XML Overview**

Prepared for: XML 2002  
December 2002

*Betty Harvey*

*[harvey@eccnet.com](mailto:harvey@eccnet.com)*

# XML Overview

Betty Harvey  
Electronic Commerce Connection  
[harvey@eccnet.com](mailto:harvey@eccnet.com)

## Table of Contents

AN GENTLE INTRODUCTION INTO XML.....	7
1.1. WHAT IS XML.....	8
1.2. XML IS NOT.....	9
1.3. XML FAMILY OF SPECIFICATIONS.....	10
1.4. BENEFITS OF XML.....	11
XML BACKGROUND.....	13
2.1. XML HISTORY.....	16
2.2. XML HISTORY.....	17
2.3. WHAT IS XML.....	18
2.4. XML DOCUMENT COMPONENTS.....	19
2.5. WHAT DOES XML LOOK LIKE.....	20
2.6. A LITTLE HISTORY OF SGML.....	21
2.7. SGML PROBLEMS.....	23
2.8. THEN CAME THE WEB.....	24
2.9. HYPERTEXT MARKUP LANGUAGE.....	25
2.10. XML ROOTS.....	26
2.11. BRAGGING ABOUT HTML.....	27
2.12. HTML PROBLEMS.....	28
2.13. POEM TAGGED WITH HTML.....	29
2.14. BRAGGING ABOUT XML 1.0.....	30
2.15. XML CRITICISMS.....	32
2.16. HTML VS. XML TAGS .....	34
2.17. XML - STRUCTURE AND CONTENT DRIVEN.....	35
WELL-FORMED DOCUMENTS.....	36
3.1. WELL-FORMED DOCUMENTS.....	37
3.2. XML WELL-FORMEDNESS.....	38
XML APPLICATION.....	39
4.1. XML DOCUMENT.....	40
4.2. XML DECLARATION.....	42

4.3. XML DOCUMENT COMPONENTS.....	43
4.4. WHAT'S IN A DTD/SCHEMA.....	44
4.5. XML ELEMENTS - STRUCTURAL BUILDING BLOCKS.....	45
4.6. WHAT IS AN ATTRIBUTE.....	46
STRENGTHS OF XML.....	47
5.1. XML IS REUSABLE.....	48
5.2. XML - REUSABLE OBJECTS.....	49
5.3. INFORMATION OBJECTS .....	50
5.4. COMMON STRUCTURES.....	51
5.5. REUSABLE INFORMATION .....	52
5.6. REUSABLE CONTENT.....	53
PRESENTATION OF XML.....	54
6.1. BENEFITS OF STANDARDIZED PRESENTATION.....	55
6.2. PRESENTATION OF THE XML.....	56
6.3. CASCADING STYLESHEET (CSS).....	58
6.4. EXTENSIBLE STYLESHEET TRANSFORMATION (XSLT).....	59
6.5. XSL-FORMATTING OBJECT (FO).....	61
6.6. ATTACHING A STYLESHEET TO XML.....	62
6.7. THE END.....	63

## Introduction

The XML Overview will provide the student a broad overview of XML. This is 45 session that will cover the following areas.

During the first section the speaker will discuss XML's background and its rich history. It will also describes the relationship of the XML specification to the ISO 8879 SGML standard and the W3C HTML specification. We will also discuss the advantages of using XML.

This second section will provide a 'gentle' introduction into XML. It will discuss what XML is and what it is not. It will also discuss some of the benefits of using XML. This section will also talk about what constitutes a *well-formed document* as defined in the W3C XML Specification.

The last section will provide a brief discussion of XML information objects and reusable components. It will demonstrate how information objects can be processed using XSLT.

## Slide 1: Introduction

- The session will cover 5 areas of XML:
  - An Gentle Introduction into XML
  - The background of XML;
  - Strengths of XML;
  - Information Objects;
  - XML Application;
  - XML Document;
  - Well-formed Document;
  - The XML Parser;
  - Structure vs. Content
  - The advantages of using XML;
  - Standards and Specifications (Navigating XML Acronym Soup)
  - How XML facilitates traditional business models and eBusiness; and
  - Briefly describe XML and legacy systems

## Slide 2: Common Acronyms Used In Tutorial

- CALSContinuous Acquisition and Lifecycle Support, Computer Aided Logistics Support, (Commerce At LightSpeed
- CSSCascading Stylesheet
- DOMDocument Object Model
- DTDDocument Type Definition
- EDIElectronic Data Interchange
- HTMLHyperText Markup Language
- ISOInternational Standards Organization
- RELAXRRegular LAnguage description for XML
- SAXSimple API for XML
- SGMLStandard Generalized Markup Language
- SOAPSimple Object Access Protocol
- STEPStandard Exchange for Product Data
- W3CWorld Wide Web Consortium - Develop specifications for the WWW.
- WWWWorld Wide Web
- WMLWireless Markup Language
- XMLeXtensible Markup Language
- XSL/XSLTXML Style Language/XML Style Language Transformation
- UTF-16 UCS Transformation Format 16 190.
- UTF-8 UCS Transformation Format 8

### Instructors Note:

There are over 250 acronyms associated with XML and XML initiatives. The above acronyms will be used in this tutorial. A list of the common acronyms are available at <http://www.eccnet.com/acronyms>.

## **Section 1: An Gentle Introduction into XML**

## Slide 3: What is XML

- XML is a generic data format
  - Describes structure
  - Describes content
  - Describes the relationship of information
- Provides a standard syntax.
- Does not provide semantics!
  - Semantics The meaning of the tags.
- Provides the ability to include business rules (metadata) information within the data.

### Instructors Note:

XML is a markup language that provides a standard syntax. It is important to understand that XML specification provides only a standard syntax. Syntax can be thought of as the construct of how the information is marked up.

XML does not provide the semantics (definition) of the elements and/or attributes used within an XML vocabulary. XML allows organizations to develop their own vocabulary for their specific application.

For example, a publishing company might use an element called <title> for a title of a book, chapter, section, etc. However, legislative bodies would create an element called <title> with a complex structure for legislation.

## Slide 4: XML is Not

- A programming language
- An automatic translation format between databases
  - Provides transformation capabilities through external processes
  - XSLTeXstensible Markup Language Transformation
  - Omnimark

### Instructors Note:

XML can be created in many different ways. Software is required to create and process of the XML. XML is an intelligent format. XML makes it easy for computers to generate and process data. XML data is also platform independent.

## Slide 5: XML Family of Specifications

XML Schema	XSL (eXtensible Style Language)
Query	XHTML
XPath	MathML
Xpointer	SMIL (Synchronized Multimedia Integration Language)
XLink	SVG (Scalable Vector Graphics)
DOM (Document Object Model)	XML Signature
RDF (Resource Description Framework)	ebXML
CSS (Cascading Style Sheets)	UDDI
RELAX (Regular Language description for XML)	

***NOTE:** Over 300 different XML related acronyms (and counting) - <http://www.eccnet.com/acronyms>*

### **Instructors Note:**

The W3C XML Specification 1.0 relates only to syntax. It does not contain information about presentation, application or semantics of the XML. Other standards and specifications have been developed around the XML 1.0 specification.

It is important to understand that there are currently hundreds of XML specifications and initiatives. Some of these specifications and initiatives are currently in direct conflict with each other.

## Slide 6: Benefits of XML

- XML provides greater levels of standardization.
  - Industry
  - Organization
  - Company
- XML has an intelligent structural framework.
- XML will eventually be used widespread on the web.
- Single, extensible vocabulary
- Application integration
  - Transport
  - Transformation
- Data aggregation
- Data validation
- Intelligent searching
- Personalization

### Instructors Note:

There are thousands of different applications using XML. XML standard vocabularies (DTD/Schema) have been developed for almost every industry.

- Legislation
- Insurance
- Airline
- Metadata
- Chemical
- Travel
- etc.

## XML Overview

---

All the benefits and more are available with XML that are contained on this file. It also allows metadata (information about the information) to be included in the XML. This provides flexible indexing and retrieval, as well as a knowledge-base for information.

## Section 2: XML Background

### Introduction

The XML background section will describe XML's rich history. It will also describe the relationship of the XML specification to the ISO 8879 SGML Standard Generalized Markup Language standard and the W3C HTML HyperText Markup Language specification.

It is important to understand where XML came from. Most people think that XML is a 'new fangled' language that has been developed since the inception of the web. In reality, XML has a long and robust history. In this section we will describe how we got XML. The 'SGML on the Web' initiative was the beginning of XML. SGML on the Web was the original vision of Yuri Rubinsky, President, SoftQuad (1952 - 1996, <http://www.utoronto.ca/atrc/rd/Rubinsky/yuri/about-yuri.html>). Charles Goldfarb is considered the 'Father of SGML' and Yuri Rubinsky is considered the "Father of SGML on the Web". Yuri worked hard to get XML as a viable standard. Seven

months after Yuri's untimely death many of Yuri's the first draft of the XML specification was released.

Yuri provided the first SGML browser for the web to the SGML community. Panorama was a free plug-in to Netscape and IE and became very popular with the SGML community.

## Slide 7: XML Background Section

- In this section we will:
  - Describe XML's rich history
  - Describe the relationship of the XML specification to the ISO 8879 SGML standard
  - Describe XML's relationship to HTML

## Slide 8: XML History

- XML is a subset of the International Standards Organization (ISO) Standard Generalized Markup Language (SGML), ISO 8879:1986
  - SGML is an ISO Standard - ISO 8879:1986
  - SGML Established Standard for 12 years.
- SGML was released as ISO 8879 in 1986
- Used in major industries
  - Manufacturing (Automobile, Heavy Equipment, Semiconductors, etc.
  - Telecommunications
  - Publishing
  - Government
  - Aviation

### Instructors Note:

If you are new to the XML world or have been working with XML for only a short time, you are probably wondering why this section is included in this section. It is important that companies who develop XML applications understand the history of XML and the importance that SGML and HTML play in the world of XML.

SGML and HTML have had a profound impact on businesses and they still play and will continue to play a significant role in development of business documents. For example, a few years ago one major on-line legal publisher claimed they had more SGML information in their database than the entire WWW has in HTML. This organization continues to use SGML as a basis for their data. Organizations that are currently using SGML have continued to use their established business practices. They are using XML publishing their information on the web.

## Slide 9: XML History

- At SGML 96 Conference, XML specification was released by a working group associated with the W3C.
- XML 1.0 is a W3C recommendation (32 pages)
  - XML became a recommendation in February, 1998

### Instructors Note:

Even with the success of Panorama, SGML was complex and many people knew that it had to be simplified before it would be accepted by the world at large. XML was born by looking at SGML and deciding what was the core functionality required. This is important: **XML is a subset of SGML!**

Some of the functionality that was extracted from XML DTD's for simplicity purposes have been put back in W3C XML Schemas.

## Slide 10: What is XML

- The eXtensible Markup Language (**Metalinguage**)
  - MetalinguageA language used to talk about language.
- A simplified subset of the Standard Generalized Markup Language (SGML)
- A standard for describing different types of data
- A standard designed to extend the use of markup languages on the WWW

### **Instructors Note:**

XML can be used to model any kind of language.

## Slide 11: XML Document Components

- Elements - Building Blocks
  - `<h1>This is an HTML/XML element</h1>`
- Attributes - Qualifiers for elements
  - `<h1 align="center">This is an HTML/XML element aligned center using the attribute 'align'</h1>`
- Entities - reusable components, links to external information, character encoding
  - `<h1>Here is a copyright "©"; character entity</h1>`
- Comments - internal comments not seen by presentation system
  - `<!--This is a comment-->`
- Processing Instructions (PI) - system specific information.
  - `<? This is a processing instruction?>`

### Instructors Note:

An XML document consists of the above components. An XML document must have at least one element. Everything else is optional. For example, the following tagged information would consist of a valid XML document.

```
<?xml version="1.0"?>
<myXML/>
```

Although this example isn't a realistic document, it shows how simplistic an XML document can be.

## Slide 12: What Does XML Look Like

```
<?xml version="1.0"?>
<poem id="poem1">
  <title>The Raven</title>
  <poet>Edgar Allan Poe</poet>
  <stanza id="stanza1">
    <line>Once upon a midnight deary, while I pondered, weak and weary,</line>
    <line>Over many a quaint and curious volume of forgotten lore-</line>
    <line>While I nodded, nearly napping, suddenly there came a tapping</line>
    <line>As of some one gently rapping, rapping at my chamber door.</line>
    <line>"`Tis some visitor," I muttered, "tapping at my chamber door-</line>
    <line>Only this and nothing more."</line>
  </stanza>
  <stanza id="stanza2">
    <line>Ah, distinctly I remember it was in the bleak December;</line>
    <line>And each separate dying ember wrought its ghost upon the floor. </line>
    <line>Eagerly I wished the morrow; - vainly I had sought to borrow </line>
    <line>From my books surcease of sorrow-sorrow for the lost Lenore-</line>
    <line>For the rare and radiant maiden whom the angels name Lenore-</line>
    <line>Nameless <whisper>here</whisper> for evermore.</line>
  </stanza>
</poem>
```

### Instructors Note:

This slides demonstrates how information can be identified according to its content. Semantic understanding can be derived from the element names. Unique ID's can also be assigned.

## Slide 13: A Little History of SGML

- Began at IBM as GML (Generalized Markup Language).
- Charles Goldfarb considered the "Father of SGML"™.
- DoD adopted SGML early as part of the CALS(Computer Aided Logistics Support aka Continuous Acquisition and Lifecycle Support aka Commerce at Lightspeed).
- Industry Standards Organizations Using SGML.
  - European Association of Aerospace Industries (AECMA)
  - Continuous Acquisition and Lifecycle Support (CALC) Defense Departments (U.S., U.K., Australia, Japan, NATO, etc.)
  - Airline Transportation Association (ATA)
  - Telecommunication Industry Forum (TCIF)
  - Railroad Industry Forum (RIF)
  - Society of Automotive Engineers (SAE)
- A lot of the same industries currently using traditional EDI.

### Instructors Note:

Standard Generalized Markup Language - ISO 8879 (SGML) is an International Standards Organization (ISO) standard. SGML became an official ISO standard since 1986. IBM had originally created an internal IBM standard called GML (Generalized Markup Language). GML was the basis for SGML. Dr. Charles Goldfarb, one of the original architects of GML at IBM continued his work to ensure that SGML became an ISO standard. Charles Goldfarb is considered the "Father of SGML" by many.

SGML was an 'industrial strength' standard. It was complicated and was very flexible. Because of the complexity and the flexibility of the standard, vendors found it very difficult to write software to support the standard. Therefore, SGML software was and is still very expensive.

However, we are seeing the costs of SGML/XML software products being driven down because of XML.

Major industries adopted SGML for their publishing standards after SGML became an ISO standard. The notable industries were:

- Government Defense Departments (U.S., Canada, U.K., Australia, Japan, to name a few)
- Manufacturing (aviation, automobile, semiconductor)
- Publishing (textbooks, medical, literary libraries). The University of Virginia and Oxford University (<http://library.ox.ac.uk/>) (two notable universities out of many) have maintain their literary collection on-line and archived in SGML.
- Telecommunications

SGML was developed to define structure and content of the information. SGML isn't concerned with the format of the information or how it is presented to potential users.

## Slide 14: SGML Problems

- High initial investment
- Complexity
- Too many options/features
- Vendors supported a subset of features
- Applications weren't portable because of various feature sets
- Lack of intuitive end-user software
  - Fear of "pointy brackets"<sup>TM</sup>\* (<>)

*NOTE: Pointy Brackets<sup>TM</sup> is a technical term!*

### Instructors Note:

Because of the complexity and high initial cost to get into the SGML market, many organizations who were looking at the technology coined a phrase SGML Sounds Good, Maybe Later. Because of the high investment only large organizations with a large IT budget could use SGML. The small and medium-sized organizations only used SGML when they were forced too by larger business partners who required information in SGML. This was true for several industries, airline manufacturers, defense departments (US, UK, Australia, etc.).



## Slide 15: Then Came The Web

- The Global Hypertext Project began in December 1990 at CERN University, European Laboratory for Particle Physics under the direction of Tim Berners Lee
- The Global Hypertext Project became to be known as the World Wide Web (WWW)
- Underlying data format for the WWW is HyperText Markup Language (HTML)

### Instructors Note:

The Global Hypertext Project began in December 1990 at CERN, European Laboratory for Particle Physics under the direction of Tim Berners Lee. The project needed a way to communicate between different buildings. HyperText Markup Language (HTML) was developed for this project. HTML was an application of SGML. HTML is an application that defines information based on its presentation. HTML information is not tagged according to its content or structure - however, it is still a SGML application because it has a defined DTD, elements and attributes.

## Slide 16: HyperText Markup Language

- HTML is an SGML application.
  - Largest SGML application in the world
  - Most successful SGML application in the world
  - Cheapest SGML application in the world
- HTML 4.0.1 released December 24, 1999 (367 pages)
- HTML specification describes the syntax and semantics of HTML.
  - XML specification only syntax (32 pages)

### Instructors Note:

HTML became the largest SGML application in the world. It became the most successful SGML application in the world. It also became the cheapest SGML application in the world. It proved that SGML could be used by the 'common man'. HTML provided the move of SGML into the mainstream of corporate computing. It was easy enough for everyone to learn. Many executives and information providers were afraid of SGML because of the learning curve and the fear of the pointy brackets. HTML was a paradigm change.

## Slide 17: XML Roots

- Yuri Rubinsky, President, SoftQuad had a vision of "SGML on the Web"™.
- Panorama was the first effort to bring a full SGML browser to the Web in 1994.

### Benefits of Panorama

- Full SGML Publishing on the Web
- Dynamic table of contents
- Easy to learn style sheet
- Support for HyTime
- Personal linking capability

### Panorama Problems

- Required DTD validation
  - Plug-in browser
  - Non-standard stylesheet
- The idea of XML came from the early SGML on the Web efforts.

### Instructors Note:

Companies and corporations flocked to HTML. However, it was soon realized by these companies that HTML was not robust enough to handle 'real business' information. SGML was still needed but SGML was still complicated. In October 1994, the second international WWW Conference in Chicago, Yuri Rubinsky, President of a small Canadian company called SoftQuad, held a session called 'SGML on the Web'. SoftQuad was (and still is) a provider of SGML/XML authoring tools. During this conference Yuri announced Panorama a browser for SGML. Panorama proved that you could provide real SGML on the Web.

Many companies who were doing SGML flocked to Panorama. One major telecommunications company bought 30,000 copies of Panorama to put on all their employees desktop for access to corporate data. The U.S. Office of the Secretary of Defense used Panorama to access administrative data.

## Slide 18: Bragging About HTML

- Cheap Lots of available tools
- ASCII editors will work
- Portable
- Easy to learn
  - Users quickly lost their fear of "pointy brackets" <>
  - Doesn't require Computer Science degree to create web pages with HTML.
- Workable and consistent hypertext facility
- Browser support

### Instructors Note:

HTML is an SGML vocabulary. HTML broke the SGML edict that SGML is about structure content and not about format. HTML tags were all about formatting. However, HTML proved to the world that you could do SGML in a cost effective environment. Cheap tools for creating HTML flooded the marketplace very quickly.

## Slide 19: HTML Problems

- Fixed formatting tags
  - XML deals with structure and content.
- No reusability or modularity
- Browser wars
- No facility to personalize
  - Not extensible
- Very little structure
  - Data relationships cannot be established.

### Instructors Note:

HTML has fixed formatting tags. You cannot use content tags to provide meaning to the HTML element. Some companies have tried to include semantic meaning for the elements by using the 'class' attributes.

There is no modularity or reusability to HTML. You can do 'server-side' includes with HTML but this is hardware/software specific.

You cannot establish relationship within HTML using the hierarchy (parent/child structure). There are basically only two structures to an HTML document, <head> and <body>.

## Slide 20: Poem Tagged With HTML

```
<HTML>
<HEAD>
  <TITLE>The Raven</TITLE>
</HEAD>
<BODY>
  <H1 ALIGN="CENTER">The Raven</H1>
  <H2 ALIGN="CENTER"> Edgar Allan Poe</H2>
  <HR>
  <P>Once upon a midnight deary, while I pondered, weak and weary,<BR>
    Over many a quaint and curious volume of forgotten lore-<BR>
    While I nodded, nearly napping, suddenly there came a tapping<BR>
    As of some one gently rapping, rapping at my chamber door.<BR>
    "'Tis some visitor," I muttered, "tapping at my chamber door-<BR>
    Only this and nothing more.
  </P>
  <P>Ah, distinctly I remember it was in the bleak December;<BR>
    And each separate dying ember wrought its ghost upon the floor. <BR>
    Eagerly I wished the morrow; - vainly I had sought to borrow <BR>
    From my books surcease of sorrow-sorrow for the lost Lenore-<BR>
    For the rare and radiant maiden whom the angels name Lenore-<BR>
    Nameless <FONT color="red">here</FONT> for evermore.
  </P>
</BODY>
</HTML>
```

### Instructors Note:

Looking at the poem above tagged in HTML, you can see that you cannot identify pieces of the poem. For example, you can't recognize the name of the poem, the author or the individual stanza's.

The lack of robust tagging limits the usability of the information. For example, with the poem tagged as HTML, you can't ask for the author of the poem 'The Raven'. You would have to do a full-text search for 'The Raven' then take your chances that you find the poem 'The Raven' and not some other reference to 'the raven'.

## Slide 21: Bragging About XML 1.0

- Identifies content according to its type not its format
- Conveys information specific to an organization or application
- Communicates this information to both humans and computers
- Works for any type of information
- All the advantages of SGML without the complexity
- Portable
- XML provides the robust functionality of SGML without most of the complex feature-set
- Vendor support, i.e., Microsoft, Netscape, IBM, Sun, Oracle, etc.
- Easy to learn
- Less expensive to implement than SGML
- Internationalization
- Web Accessibility
- Easy to build applications

### Instructors Note:

XML 1.0 provides most of the advantages of SGML without the complexity.

It is important to distinguish the difference between all the specifications around XML and XML 1.0. The surrounding specifications, XML Schema, RDF, XSL(T), can be difficult to understand. However, the core XML is very easy.

It doesn't take the large cost investment to use XML that it did with SGML. A lot of good XML tools are open source. It is inexpensive to create and present XML information. There are many times more tools

available for XML than there were for SGML. SGML tools were very limited.

## Slide 22: XML Criticisms

- A lot of hype (*Hype is dying down and reality is setting in*)
- Hard to distinguish reality from hype

### **Complicated specifications**

- W3C Schema Specification
- ebXML
- RDF
- *and more ...*

### **Conflicting Specifications**

- Schema (W3C/Relax NG)
- Repository (ebXML/UDDI)
- Transport (Simple Object Access Protocol [SOAP]/ebXML Transport and Routing Protocol [TRP])
- Stylesheet
  - eXtensible Style Language (XSL)
  - Cascading Stylesheet (CSS)
  - Document Style and Semantic Stylesheet Language (DSSSL)
- Confusion about which rules-based specification to use
  - DTD versus Schema
- Browser implementations slow
  - Browser incompatibility (wars)

### **Instructors Note:**

As noted before, the activities currently around XML are difficult to navigate. It is important that managers and developers understand the implications of any XML development before trudging ahead.

In some cases, the specification that is used for a particular project, i.e., SOAP/ebXML TRP, Schema(s)/DTD, may depend on the software that will be used to implement the XML project.

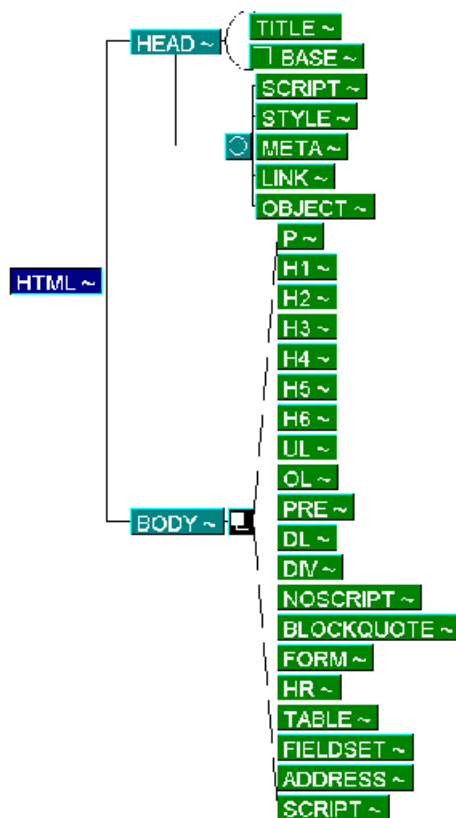
A good example would be an authoring tool. Currently, authoring tools (Arbortext Epic, WordPerfect, Frame +SGML) do not support schemas. Corel's XMetaL 3.0 currently supports W3C Schema language. Epic editor is rumored to support schemas during a future release.

## Slide 23: HTML vs. XML tags

### HTML

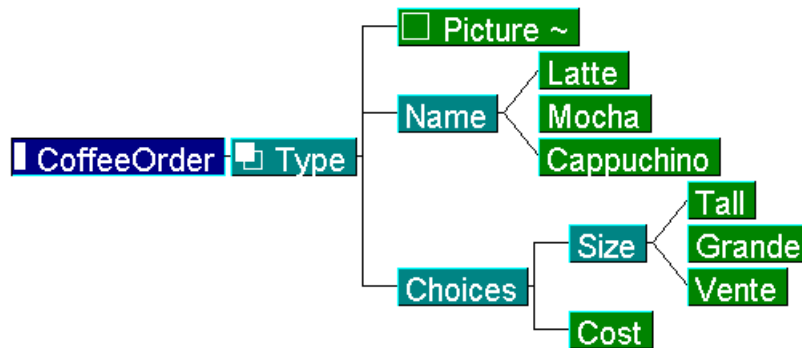
- Format Driven
- Little Structure
- Mostly Formatting Tags
- No Intelligence in the Data

### Hierarchical Diagram of HTML DTD



## Slide 24: XML - Structure and Content Driven

### Hierarchical Diagram of a Coffee Order DTD



### Instructors Note:

Comparing the two hierarchical graphics above, you can see that HTML has two levels of hierarchy, `<head>` and `<body>`. The `<body>` can contain any element in any order.

The `<CoffeeOrder>` element shows that you can order multiple types of `<Latte>`, `<Mocha>` or `<Cappuchino>`. You can also see that the model is extensible if the menu grows, for example if they decide to include an `<ExtraGrande>` size.

## Section 3: Well-formed Documents

### Introduction

There are two types of XML, well-formed and valid. Well-formed documents are the least stringent type. A well-formed document simply requires that all elements are cleanly nested. Also, all attribute values must be enclosed in quotes ("..." or '...').

Valid documents, on the other hand, must include a DTD or a Schema and adhere to it!

## Slide 25: Well-Formed Documents

- Contains one or more elements.
- There is exactly one element, called the **root element**, or document element.
- Each element has a start tag.
- Each element has an end tag.
- Each attribute value is delimited using quotes (single or double).
- Element and attribute names are case sensitive, i.e., <p> and <P> are considered two separate and distinct elements.

### Instructors Note:

A well-formed document requires that all elements must have a start tag and an end tag. It also requires each attribute value to be delimited in quotes (double or single are fine as long as they are consistent).

If you are familiar with HTML authoring, this is different. In HTML, you can have a start tag with an ending tag. Each attribute has to be enclosed in quotes.

## Slide 26: XML Well-Formedness

### Valid HTML fragment

```
<H1 align=center>HTML snippet</h1>
<hr>
<p>This is a paragraph
<p>Next paragraph
```

- Start tag for *H1* is upper case whereas the end tag is lower case
- The attribute value is not enclosed in quotes.
- The `<hr>` element is an empty element so it doesn't have content or a closing tag.
- The `<p>` elements do not have an end tag

Below is the resulting well-formed document fragment.

### Well-formed fragment

```
<h1 align="center">HTML snippet</h1>
<hr/>
<p>This is a paragraph</p>
<p>Next paragraph</p>
```

### Instructors Note:

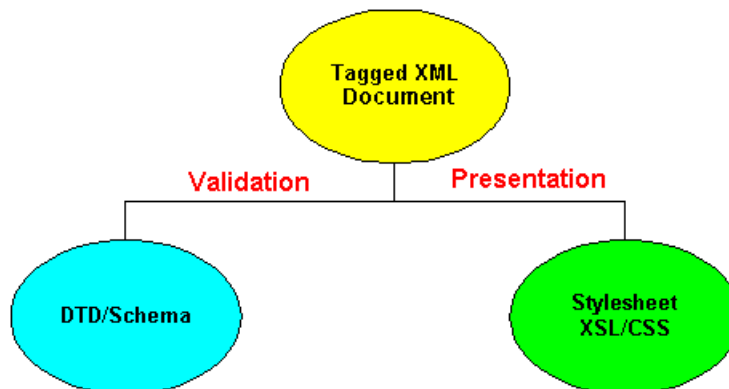
Although it is good form to enclose attribute values in quotes in attribute values, HTML browsers are lenient in this respect.

## **Section 4: XML Application**

## Slide 27: XML Document

- An XML document is composed of four components
  - XML Declaration - `<?xml version="1.0" encoding="UTF-8"?>`
    - The declaration is
  - Document Type Definition (DTD) or Schema
    - The Rules that define an XML
  - Tagged document (instance)
  - Stylesheet
    - eXtensible Stylesheet (XSL)
    - Cascading Stylesheet
    - Document Style Semantic and Specification Language (DSSSL)

### XML Document



### Instructors Note:

A complete XML document is comprised of three components. The DTD or Schema is used to create a rule-based document. It also is used to validate the document to ensure that has been created based upon the defined rules.

The stylesheet is used to present the document in a human-readable format. A browser or publishing engine uses the stylesheet to represent the document visually.

## Slide 28: XML Declaration

- Three parts:
  - version number - required
  - encoding - optional
  - standalone-optional
- Can be used with both SGML and XML tools. Available from:

<http://www.w3.org/TR/NOTE-sgml-xml.html>

### Simple XML Declaration

```
<?xml version="1.0"?>
```

### Full XML Declaration

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

### Instructors Note:

The version number is the only required component of the XML declaration. Currently the version is 1.0.

The encoding declaration describes what character encoding is used.

The standalone declaration declares what components of the document type definition are necessary for complete processing.

## Slide 29: XML Document Components

- Elements - building blocks
- Attributes
  - qualifiers of elements
  - properties of elements
- Entities
  - Reusable components
  - References to binary information
  - Special character handling

## Slide 30: What's in a DTD/Schema

- The rules in the DTD describe:
  - the names of allowable elements (tags)
  - the content of each element type
  - the structure of the document, including:
    - the order in which elements must appear
    - how often the elements can appear
  - the properties of the elements (which are called attributes)

### Example DTD

```
<!ELEMENT poem (title, poet, stanza+)>
<!ATTLIST poem
    id      ID      #REQUIRED>

<!ELEMENT title (#PCDATA)>

<!ELEMENT poet (#PCDATA)>

<!ELEMENT stanza (line+)>
<!ATTLIST stanza
    id      ID      #REQUIRED>

<!ELEMENT line  (#PCDATA | whisper)*>

<!ELEMENT whisper (#PCDATA)>
```

## Slide 31: XML Elements - Structural Building Blocks

- The DTD describes:
  - What elements are allowed
  - How the elements are related
  - The allowable content of an element
  - The properties (attributes) of the element
  - Elements have unique names and lengths are not restricted
  - The first NAME character must be a letter, “\_” or “:”

## Slide 32: What is an Attribute

An attribute is a property that is associated to an element.

- Examples:
  - Unique identifier
  - Revision level of a document
  - Review status of a proposal
  - Author of a review comment
  - Size of a graph
  - Internal audit designation of a repair manual

## **Section 5: Strengths of XML**

### **Introduction**

During this section we will outline why you would want to use XML. We will also talk about the capabilities of using XML.

## Slide 33: XML is Reusable

- Information can be reused via addressing mechanisms.
- Information can be reused by external entities (reusable modules, i.e., boilerplate text).
- XML is currently being expanded to allow more robust addressing mechanisms (XLink/Xpointer)

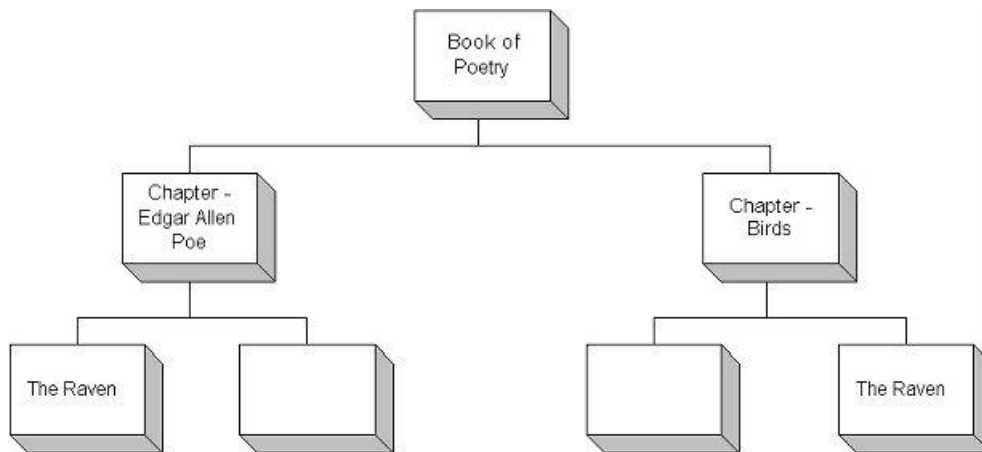
### Instructors Note:

XML provides the powerful capability of reusing information objects. An information object is a logical piece of information. A good example of an information object is an address. An individual address can be used in many different contexts within a single document, transaction or message. For instance, the same address could be reused within a purchase order for requestor, shipping address, etc. This address can also be used in multiple transactions. XML provides the facility for these reusable objects. *Create once - reuse many.*

## Slide 34: XML - Reusable Objects

Showing Hierarchical Relationships

### Reusable Object



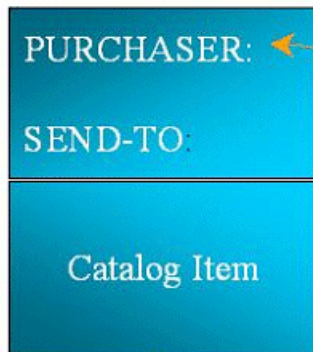
### Instructors Note:

The graphic above, shows how an information object can be reused. For example, the poem "The Raven" by Edgar Allen Poe can be used in multiple locations, one chapter devoted to Edgar Allen Poe and another in a chapter concerning birds in poetry. This becomes really useful for on-line information. Everywhere that the poem "The Raven" could be referenced you would have only one source of the document.

## Slide 35: Information Objects

### Information Objects

#### Purchase Order DTD



#### Invoice DTD



#### Catalog DTD



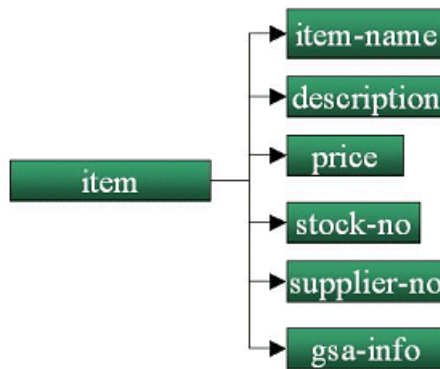
#### Instructors Note:

This graphic shows how the information objects for *purchaser* can be used in both an invoice and a purchase order. It also shows how the *catalog item* can be used in the catalog, invoice and purchase order. In the following slides we will show how the information objects remain the same, even though the objects are processed differently depending on the particular document model (invoice, purchaser and catalog item).

## Slide 36: Common Structures

- Common structures (information objects) allow you to:
  - Use common structures across multiple information components.
  - Reuse common fragments organization or industry

### Example of a Common Business Object



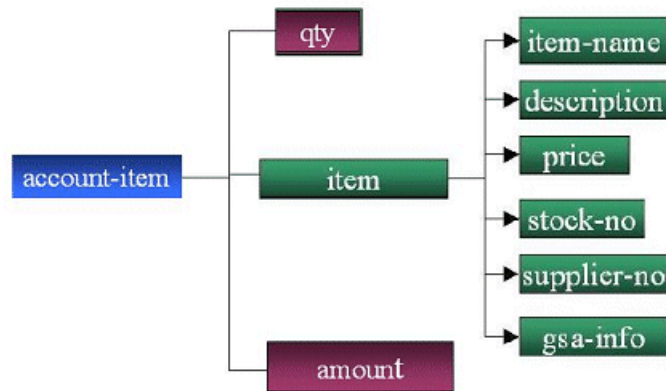
### Instructors Note:

The above model shows how a common information object can be created for a standard structure. This model can be used in multiple contexts, as well as multiple documents.

An example of a common information object would be a table. There are currently two table models in wide use. The first and oldest model is the CALS (Computer Aided Logistics Support) table model. The second is the HTML table model. The CALS table model is the more robust of the two. However, both table models have been widely used in XML applications. In some cases, the table models have been enhanced to allow for individualized tables, i.e., source notes, etc.

## Slide 37: Reusable Information

### Common Business Object Is Reusable



- Example of Catalog
- Example of Purchase Order
- Example of Invoice

### Instructors Note:

The graphic above shows how the information object `<item>` can be enhanced for the purchase order and invoice applications by wrapping the `<item>` element in a parent element and adding the `<qty>` element before the item and the `<amount>` after the item. This way, an external program can be used to calculate the `<amount>` element once the `<qty>` ordered is known.

## Slide 38: Reusable Content

- Reusable content enables
  - Author once - reuse many.
  - Authoritative authoring.
  - Data consistency.

### Instructors Note:

This slide discuss the advantages of having reusable content, as well as reusable content models. Information can be created once and reused as many times as possible.

It also provides *authoritative authoring*. For example, warnings, cautions and notes in technical documentation is prescribed by convention and sometimes law. Organizations usually have a governing organization that created and manages warning, caution and notes verbage and usage. XML provides the capability of having one department create the warnings, cautions and notes and the rest of the organization use them without modification.

## Section 6: Presentation of XML

## **Slide 39: Benefits of Standardized Presentation**

- The same stylesheets can work across all platforms
- One stylesheet language can be used for a class of documents
- Stylesheet code can be re-used for different document types
- Many different applications can process stylesheets that use the same standard

### **Instructors Note:**

Presentation standards have been developed for presenting XML. This provides the ability to create one stylesheet and present it across all platforms. There are currently two stylesheet specifications that can be used with XML (CSS and XSL/XSLT).

## Slide 40: Presentation of the XML

- Presentation of XML based on XML tags.
- Provides flexibility to other formats, HTML, CD-ROM, paper, etc.
- Standardized Stylesheets
  - Cascading StyleSheet (CSS)
  - XML StyleSheet Transformation (XSLT)
  - XML StyleSheet (XSL)



[Link to Coffee Example.](#)

```
<!DOCTYPE CoffeeOrder SYSTEM "coffee.dtd">
  <CoffeeOrder>
    <Type>
      <Name> <Latte/> </Name>
      <Size> <Grande/> </Size>
      <Cost>$3.40</Cost>
    </Type>
    <Type>
      <Name> <Mocha/> </Name>
      <Size> <Vente/> </Size>
      <Cost>$4.40</Cost>
    </Type>
    <Type>
      <Name> <Cappuchino/> </Name>
      <Size> <Tall/> </Size>
      <Cost>$2.40</Cost>
    </Type>
  </CoffeeOrder>
```

### Instructors Note:

Presentation of the XML is based on the tag elements. If you look at the tagged example, you will see that there is no form elements used, there isn't any presentation information included in the text. However, showing the example shows a nice presentation order form based on

the XML elements. The presentation can also be attached to attributes as well.

## Slide 41: Cascading Stylesheet (CSS)

- W3C Recommended Specification - May 12, 1998
- Support for CSS2 in both MS IE 5/6 and Netscape 5.
- Doesn't require transformation of XML (down translation).

```
P {  
  font-family: non-serif;  
  font-size: medium;  
  color: black;  
  overflow: visible;  
  margin-left: 40px;  
  margin-right: 40px;  
}
```

CSS stylesheets can be used with HTML and XML. CSS can also be used to enhance HTML presentation with XSLT transformations to HTML.

## Slide 42: eXtensible Stylesheet Transformation (XSLT)

- XSL Transformations (XSLT) - Version 1.0
- W3C Proposed Recommendation
- Describes syntax and semantics for transforming
- XML documents into other XML documents.
  - HTML
  - WML
  - XML
- XSL Parsers
  - XT by James Clark ([www.jclark.com](http://www.jclark.com))
  - MSXSL - Microsoft
  - SAXON - Michael Kaye
  - Cocoon - ([www.apache.org](http://www.apache.org))

```
<xsl:template match="CoffeeOrder">
  <html>
    <head>
      <title>Cool Coffee Menu </title>
    </head>
    <body font-family="Arial, helvetica, sans-serif"
          font-size="10pt" bgcolor="#EEEEEE">
      <h1 align="center">Cool Coffee Menu</h1>
      <hr width="75%" />
      <form method="POST">
        <center>
          <table border="1" cellpadding="10">
            <tr>
              <th colspan="8" align="center">Place Your Order</th>
            </tr>
            <xsl:apply-templates/>
          </table>
        </center>
        <p align="center">
          <a href="order.xml">Place and Order</a>
        </p>
      </form>
    </body>
  </html>
</xsl:template>
```

### Instructors Note:

XSLT defines how to transform XML documents into other XML documents or into HTML or text documents. The XSLT specification defines how to filter, sort and transform an XML information which allows presentation to be applied to the transformed document.

## Slide 43: XSL-Formatting Object (FO)

- Part of the W3C XSL Specification
- Draws from XSLT and CSS Specifications
- Provides formatting capability for Print
- XSL-FO Processors
  - RenderX ([www.renderx.com](http://www.renderx.com))
  - Antenna House ([www.antennahouse.com](http://www.antennahouse.com))
  - Apache FOP (Formatting Object Processor)

```
<xsl:template match="para">
  <fo:block
    font-family="Times"
    font-size="11pt"
    margin-left="25pt"
    margin-right="25pt"
    space-before.minimum="18pt">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>
```

### Instructors Note:

XSL-FO is the style component of the XSL specification. The XSL-FO specification provides the mechanism to support complex page layout, similar to desktop publishing capability. It uses CSS as a basis for the style attributes. It also used XSLT as the mechanism to transform an XML document into an XSL-FO document.

If you are reading these instructor notes, the printed copies have been created using XSL-FO for formatting.

## Slide 44: Attaching a Stylesheet to XML

- Stylesheets are attached to XML files via a processing instruction.
- A specification has not been approved yet for attaching stylesheets.
- De Facto approach
- Two Mime Types (text/xsl and text/css)
- href attribute can be a URL/URI

```
<?xml-stylesheet href="poem.xsl" type="text/xsl"?>  
<?xml-stylesheet href="poem.css" type="text/css"?>
```

### Instructors Note:

You can attach the stylesheet to an XML document using a processing instruction. Currently there isn't a definitive standard for attaching stylesheets. However, browsers recognize this mechanism.

**Slide 45: THE END**