Metadata Standards and Applications

4. Metadata Syntaxes and Containers
Goals of Session

- Understand the origin of and differences between the various syntaxes used for encoding information, including HTML, XML and RDF
- Discover how container formats are used for managing digital resources and their metadata
Overview of Syntaxes

- **HTML, XHTML**: Hypertext Markup Language; eXtensible Hypertext Markup Language
- **XML**: Extensible Markup Language
- **RDF**: Resource Description Framework
HTML

- HyperText Markup Language
- HTML 4 is the current standard
- HTML is an SGML (Standard Generalized Markup Language) application conforming to International Standard ISO 8879
- Widely regarded as the standard publishing language of the World Wide Web
- HTML addressed the problem of SGML complexity by specifying a small set of structural and semantic tags suitable for authoring relatively simple documents
XHTML

- XML-ized version of HTML 4.0, tightens up HTML to match XML syntax
  - Requires ending tags, quoted attributes, lower case, etc., to conform to XML requirements

- XHTML is a W3C specification, redefining HTML as an XML implementation, rather than an SGML implementation

- Imposes requirements that are intended to lead to more well-formed, valid XML, easier for browsers to handle
An XHTML Example

Metadata Standards & Applications
XML

- Extensible Markup Language
- A ‘metamarkup’ language: has no fixed tags or elements
- Strict grammar imposes structure designed to be read by machines
- Two levels of conformance:
  - well-formed--conforms to general grammar rules
  - valid--conforms to particular XML schema or DTD (document type definition)
XML is the *lingua franca* of the Web

- Web pages increasingly use at least XHTML
- Business use for data exchange/messaging
- Family of technologies can be leveraged
  - XML Schema, XSLT, XPath, and XQuery
- Software tools widely available (many open source)
  - Storage, editing, parsing, validating, transforming and publishing XML
- Microsoft Office 2003 supports XML as document format (WordML and ExcelML)
- Web 2.0 applications are based on XML
An XML Schema May Define:

- What elements may be used
- Of which types
- Any attributes
- In which order
- Optional or compulsory
- Repeatability
- Sub-elements
- Logic
Anatomy of an XML Record

- **XML declaration**--prepares the processor to work with the document

- **Namespaces** (uses `xmlns:prefix` and a URI to attach a prefix to each element and attribute)
  - Distinguishes between elements and attributes from different vocabularies that might share a name (but not necessarily a definition) using association with URIs
  - Groups all related elements from an application so software can deal with them
  - The URIs are the standardized bit, not the prefix, and they don’t necessarily lead anywhere useful, even if they look like URLs
XML Anatomy Lesson

<marc:subfield code="a">Metadata in practice</marc:subfield>
Namespace Anatomy Lesson

XML Namespace

xmlns:dc="http://purl.org/dc/elements/1.1/"

Namespace Identifier

Namespace Prefix
RDF

- Resource Description Framework--A language for describing resources for the web
- Structure based on “triples”
- Focused on exchange of information between different kinds of organizations and usages
- Considered an essential part of the Semantic Web
- Can be expressed using XML
Some RDF Concepts

- **A Resource** is anything that you want to describe; it’s most often identified with a URI, such as:
  
  http://dublincore.org/documents/usageguide/

- **A Class** is a category; it is a set that comprises individuals

- **A Property** is a Resource that has a name, such as "creator" or "homepage"

- **A Property value** is the value of a Property, such as "George Washington" or "http://dublincore.org" (note that a property value can be another resource)
The combination of a **Resource**, a **Property**, and a **Property value** forms a **Statement** (includes a subject, predicate and object)

An example **Statement**: "The editor of [http://dublincore.org/documents/usageguide/](http://dublincore.org/documents/usageguide/) is Diane Hillmann"

- The subject of the statement above is: [http://dublincore.org/documents/usageguide/](http://dublincore.org/documents/usageguide/)
- The predicate is: editor
- The object is: Diane Hillmann
RDF and OWL

- RDF does not have the language to specify all relationships
- Web Ontology Language (OWL) can specify richer relationships, such as equivalence, inverse, unique
- RDF and OWL may be used together
- Resource Description Framework Schema (RDFS): a syntax for expressing relationships between elements
An XML/RDF Example

```xml
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://www.dlib.org">
    <dc:description>The D-Lib program supports the community of people with research interests in digital libraries and electronic publishing.</dc:description>
    <dc:publisher>Corporation For National Research Initiatives</dc:publisher>
    <dc:date>1995-01-07</dc:date>
    <dc:subject>
      <rdf:Bag>
        <rdf:li>Research; statistical methods</rdf:li>
        <rdf:li>Education, research, related topics</rdf:li>
        <rdf:li>Library use Studies</rdf:li>
      </rdf:Bag>
    </dc:subject>
    <dc:type>World Wide Web Home Page</dc:type>
    <dc:format>text/html</dc:format>
    <dc:language>en</dc:language>
  </rdf:Description>
</rdf:RDF>
```
Overview of Container Formats

- A container format is used to package together all forms of metadata and digital content
- Use of a container is compatible with, and an implementation of, the OAIS information package concept
- METS: packages metadata with objects or links to objects and defines structural relationships
- MPEG 21 DID: represents digital objects
 Metadata Encoding & Transmission Standard
Developed by the Digital Library Federation, maintained by the Library of Congress
“... an XML document format for encoding metadata necessary for both management of digital library objects within a repository and exchange of such objects between repositories (or between repositories and their users).”
METS is open source and developed by open discussion
Cultural heritage community is the main audience
METS Usage

- To package metadata with digital object in XML syntax
- For retrieving, storing, preserving, serving resource
- For interchange of digital objects with their metadata
- As an information package in a digital repository (may be a unit of storage or a transmission format)
METS Sections

- Defined in METS schema for navigation & browsing
  - 1. Header (XML Namespaces)
  - 2. File inventory
  - 3. Structural Map & Links
  - 4. Descriptive Metadata (not part of METS but uses an externally developed descriptive metadata standard, e.g. DC, MODS)
  - 5. Administrative Metadata (points to external schemas):
    - 1. Technical, Source
    - 2. Digital Provenance
    - 3. Rights
The structure of a METS file

- METS
  - fileSec -> file inventory
  - dmdSec -> descriptive metadata
  - amdSec -> administrative metadata
  - behaviorSec -> behaviour metadata
  - structMap -> structural map
METS Extension Schemas

“Wrappers” or “sockets” where elements from other schemas can be plugged in
- Uses the XML Schema facility for combining vocabularies from different Namespaces

Endorsed extension schemas:
- Descriptive: DC, MODS, MARCXML
- Technical metadata: MIX (image); textMD (text)
- Preservation related: PREMIS
MPEG-21 Digital Item Declaration (DID)

- ISO/IEC 21000-2: Digital Item Declaration
  - An alternative to represent Digital Objects
  - Supported by some repositories, e.g., aDORe, DSpace, Fedora
- Model that represents compound objects (recursive “item”)
- MPEG DID is an ISO standard and has industry support, but it is often implemented in a proprietary environment and the standards development is closed (as is ISO in general)
MPEG 21 Abstract Model

**container**: grouping of *items* and descriptor/statement constructs pertaining to the container

**item**: represents a Digital Item aka Digital Object aka asset. **Descriptor/statement** constructs convey information about the Digital Item

**component**: binding of descriptor/statements to datastreams

**resource**: datastream
An Exercise

- Encode a simple resource in both DC and MARC using XML
- Use the template forms provided