Semantic Web

Lin Zuoquan

Information Science Department

Peking University

lz@is.pku.edu.cn

http://www.is.pku.edu.cn/~lz/teaching/stm/saswws.html

Courtesy some graphic slides from online

Outline

- Overview
- Semantics
- XML
- Metadata
- Ontology

Overview

- The Web
- The Semantic Web

The Web

- **Internet.** Virtually of the computers in the world have been connected.
  - **Scale.** Every day many more computing and communication devices are joining.
  - **Power.** Raw computing power continues to climb.
  - **Wireless.** New technologies are creating a pervasive, ubiquitous computing environment
- **Web.** The World Wide Web is a set of globally connected nodes.
  - anyone can publish content and provide services, powerful search engines support discovery, evolving standards enhance interoperability.
  - each node has a URL (uniform resource locator).

The Problems

- The Web is so simple to use that it is easy to get lost.
- Information is huge while information description (metadata) is poor.
- Search results by the keyword based search engines are not preciseand efficient.

The way we will be...

People, agents, devices, & services need to

- Find others in their environment
- Describe the services they offer and seek
- Exchange APIs
- Negotiate for services, permissions, privacy, payment, ...
- Reason about services to create composite services
- Coordinate and cooperate as needed
- Sense their context and the activities of humans
- Deal with new entities never before encountered

And to do this dynamically

HTML

- HTML provides only linear links and contains very poor semantics.
- Could we tell the machine what the different parts of the text represent?

XML in focus

- XML ≠ machine accessible meaning

The Web in three generations

1 Hand-coded (HTML) Web content

- easy access through uniform interface
- huge authoring and maintenance effort
- hard to deal with dynamically changing content

2 Automated on-the fly content generation

- based on templates filled with database content
- later extended with XML document transformations

3 Automated processing of content

- The Semantic Web (SW)

The Semantic Web

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

__ Berners-Lee, Hendler and Lassila, The Semantic Web, Scientific American, 2001

Tim Berners Lee
Semantic Web

"In communicating between people using the Web, computers and networks have as their job to enable the information space, and otherwise get out of the way. But doesn't it make sense to also bring computers more into the action? – part two of the dream!

"… This creates what I call a Semantic Web – a web of data that can be processed directly or indirectly by machines."

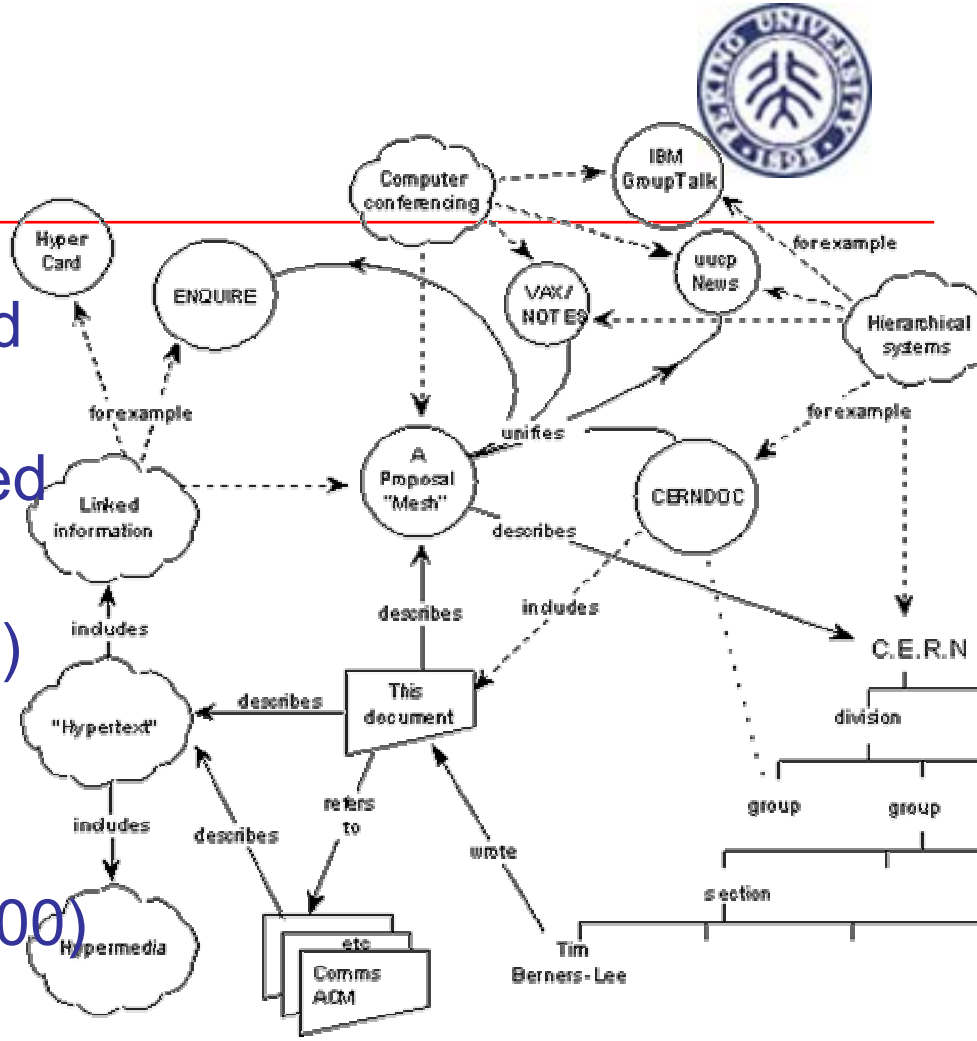
<<Weaving the Web>>, pp 177.

Web vs. SW

- **Whereas the Web has made people smarter, the SW will make machines smarter.**
- **The current Web *stores* things whereas the SW enables agents which *do* things.**

History

- Tim Berners-Lee proposed WWW as a Web of relationships among named objects (89)
- Guha designed MCF (~94)
- XML+MCF=>RDF (~96)
- RDF+OO=>RDFS (~99)
- RDFS+KR=>DAML+OIL (00)
- W3C's SW activity (01)
- W3C's OWL (02)



http://www.w3.org/History/1989/proposal.ht

SW in progress

- There are important language aspects which need more work
- Many tools need to be created
- Applications need to be explored
- The W3C is developing a new SW language
  - OWL: Ontology Web Language
- SW ideas will migrate into other standards (e.g., basic XML, WSDL, .NET)

SW Principles

- Everything have URIs on the web
- Partial information is tolerated
- Trust models are critical
- Information evolution is supported
- Minimalist design
  - Make the simple things simple, and the complex things possible. Standardize no more than is necessary.
- Common data model
  - To support interoperability and knowledge sharing

Adapted from Eric Miller, W3C

Semantic Web *does* what?

- **Concept-based search**

  ≠ keyword-based search
- **Semantic navigation**

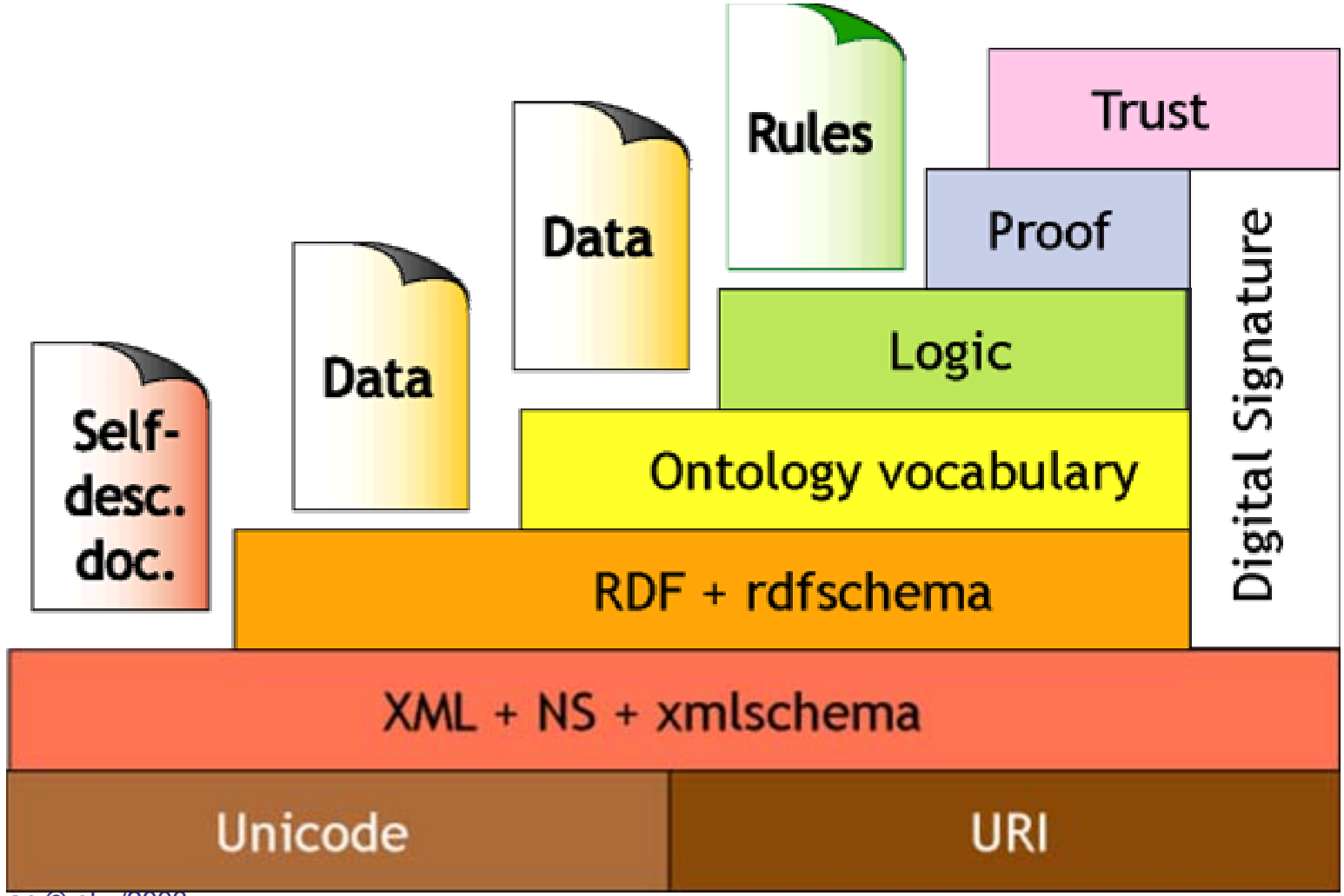
  ≠ link-based navigation
- **Personalization**

  ≠ one size fits all
- **Query answering**

  ≠ document retrieval
- **Services**

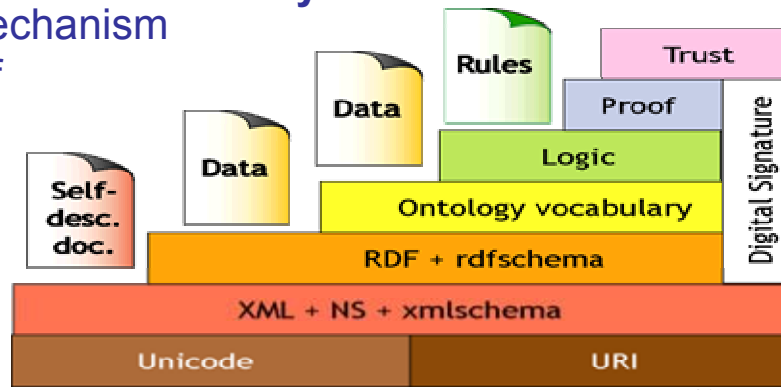
  ≠ CGI calls, but service-description languages, negotiation, service composition, etc

SW Layers


Rules
Trust
Data
Proof
Data
Logic
Self-desc. doc.
Ontology vocabulary
Digital Signature
RDF + rdfschema
XML + NS + xmlschema
Unicode
URI

SW Layers(contd.)

- The **Unicode and URI layers** make sure that we use international characters sets and provide means for identifying the objects in Semantic Web.
- The **XML layer with namespace(NS) and schema definitions** make sure we can integrate the Semantic Web definitions with the other XML based standards.
- With **RDF and RDFSchema[RDFS]** it is possible to make statements about objects with URI's and define vocabularies that can be referred to by URI's. This is the layer where we can give types to resources and links.
- The **Ontology layer** supports the evolution of vocabularies as it can define relations between the different concepts.
- The **Digital Signature layer** is used for detecting alterations to documents.
- The **top layers: Logic, Proof and Trust**, are currently being researched and simple application demonstrations are being constructed.
- The **Logic layer** enables the writing of rules while the **Proof layer** executes the rules and evaluates together with the **Trust layer** mechanism for applications whether to trust the given proof or not.



SW Applications

- Search engines
- Browsing on-line stores (B2C)
- Service description and integration (B2B)
- Tailored multimedia information

Problems with current search engines

- Current search engines = keywords:
  - high recall, low precision
  - sensitive to vocabulary
  - insensitive to implicit content

Search engines on SW

- concept search instead of keyword search
- semantic narrowing/widening of queries
- query-answering over >1 document
- document transformation operators

Problems with current B2C

- manual browsing is time-consuming and inefficient
- every shopbot requires a series of wrappers
  - work only partially
  - extract only explicit information
  - must be updated frequently

B2C on SW

- Software agents "understand" product descriptions
  - enabling automatic browsing
- Procedural wrapper-coding becomes declarative ontology-mapping
  - improving robustness and simplifying maintenance

B2B on SW

- Semantic Web technology is
  - Cheaper
  - Flexible
  - Integrated with "document" Web
- Provides interoperable semantics for
  - vertical markets (verticalnet.com)
  - horizontal markets

More Applications

- Web Portal
- Web Agents
- Multimedia Collections
- Web Management
- Design Documentation
- Ubiquitous computing
- And more

Semantics

- What's Semantics
- Formal Semantics
- Semantics on Web

Big Challenges

- Semantics are critical to support the next generation of the Web.
- The important contribution of the Semantic Web, vis-à-vis the current Web, is the ability to represent and process descriptions of every resource on the Web.
- A resource description, informally called its “semantics”, includes that information about the resource that can be used by machines - not just for display purposes, but for using it in various applications.

What's Semantics

- Syntax
- Semantics
- Pragmatics

Formal Semantics

- Tarski's Semantics
- Possible World Semantics
- Operational Semantics

Semantics on Web

- Informal Usage
- Semantics in SW

XML

- XML in brief
- Limitations for Semantic Markup

XML in brief

- XML (eXtensible Markup Language) is a meta(-markup) language for text documents
- XML is a structural and semantic markup language, not a presentation language (as HTML)
- XML is a cross-platform, software and hardware independent tool for transmitting information
- XML is used to create structured, self-describing documents that conform to a set of rules (*grammar*) created for each specific language
- Documents that satisfy the grammar are said to be *well-formed*
- The markup permitted in a particular XML document can be declared in a Document Type Definition (DTD,or XML Schema). Documents that match the DTD are said to be *valid*

XML History

- XML was developed by an XML Working Group (originally the SGML Editorial Review Board) formed under the auspices of W3C in 1996. It was chaired by Jon Bosak (Sun) with the active participation of an XML Special Interest Group (previously the SGML Working Group) also organized by the W3C.
- The design goals for XML are:
  - **XML shall be straightforwardly usable over the Internet.**
  - **XML shall support a wide variety of applications.**
  - **XML shall be compatible with SGML.**
  - **It shall be easy to write programs which process XML documents.**
  - **The number of optional features in XML is to be kept to the absolute minimum, ideally zero.**
  - **XML documents should be human-legible and reasonably clear.**
  - **The XML design should be prepared quickly.**
  - **The design of XML shall be formal and concise.**
  - **XML documents shall be easy to create.**
  - **Terseness in XML markup is of minimal importance.**
- This specification, together with associated standards (Unicode and ISO/IEC 10646 for characters, Internet RFC 1766 for language identification tags, ISO 639 for language name codes, and ISO 3166 for country name codes), provides all the information necessary to understand XML Version 1.0 and construct computer programs to process it.

Tag Language

- Using tags to describe a part or the whole content to indicate how the part or the content is presented, structured, or meant.
- XML is a subset borrowed from SGML
  - SGML with 80% functionality but 20% complexity
- By name
  - it is a markup language - marking up content:
    - <mark>Content</mark>
  - it is extensible - user can define whatever they like:
    - <my_name>My Name</my_name>

XML example

```
<?xml version="1.0" encoding="UTF-8">
<person born='23/06/1912' died='07/06/1954'>
    <name>
        Alan Turing
    </name>
    <profession>
        mathematician and computer scientist
    </profession>
    <profession>
        cryptographer
    </profession>
</person>
```

XML vs. HTML

- General Structure:
  - Both have Start tags and end tags.
- Tag Sets:
  - HTML has set tags
  - XML lets you create your own tags.
- General Purposes:
  - HTML focuses on structure and presentation -- "look and feel"
  - XML focuses on the structure of the data, not presentation
- XML is *not* meant to be a replacement for HTML. In fact, they are usually used together.

Book Catalog in HTML

```
<HTML>
<BODY>
<H1>Harry Potter</H1>
<H2>J. K. Rowling</H2>
<H3>1999</H3>
<H3>Scholastic</H3>
</BODY>
</HTML>
```

HTML conveys the "look and feel" of your page.

As a human, it is easy to pick out the publisher.

But, how would a computer pick out the publisher?

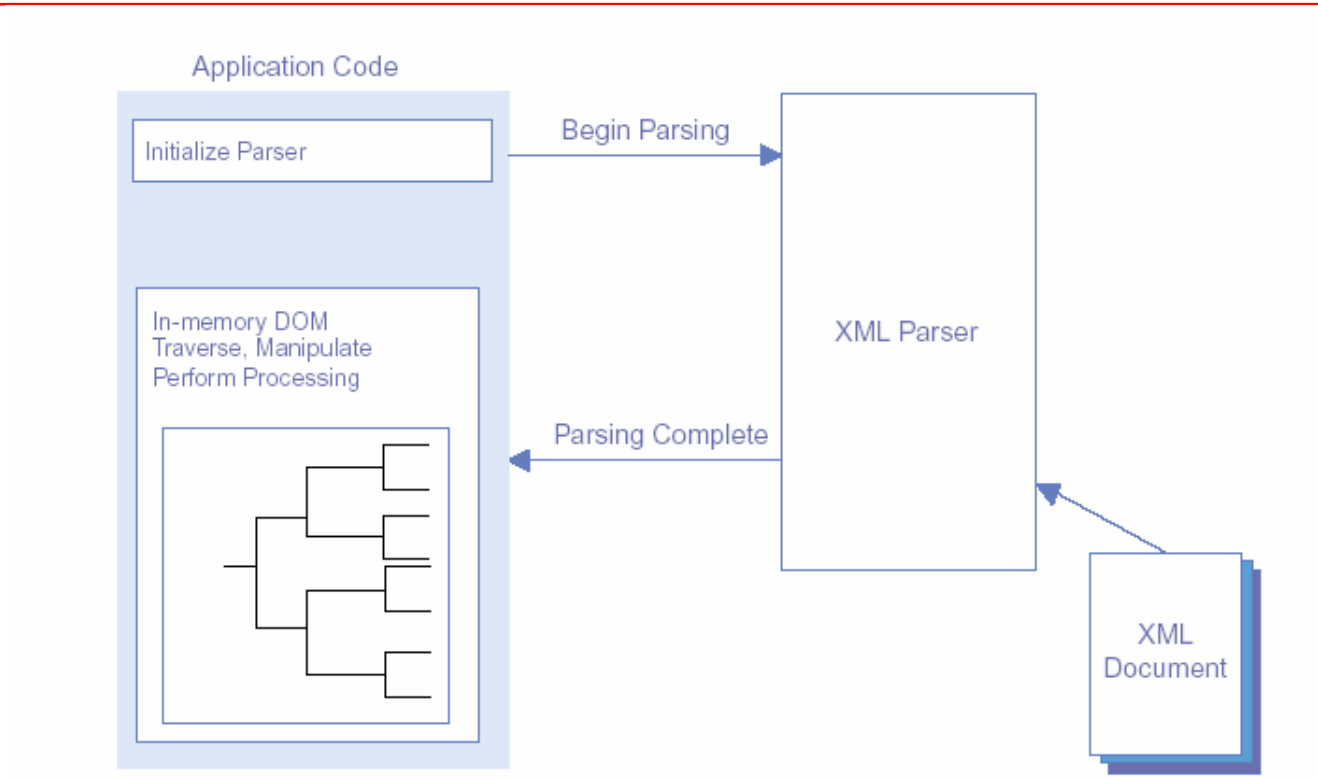
Answer: XML

Book Catalog in XML

```
<BOOK>
   <TITLE>Harry Potter</TITLE>
   <AUTHOR>J. K. Rowling</AUTHOR>
   <DATE>1999</DATE>
   <PUBLISHER>Scholastic</PUBLISHER>
</BOOK>
```

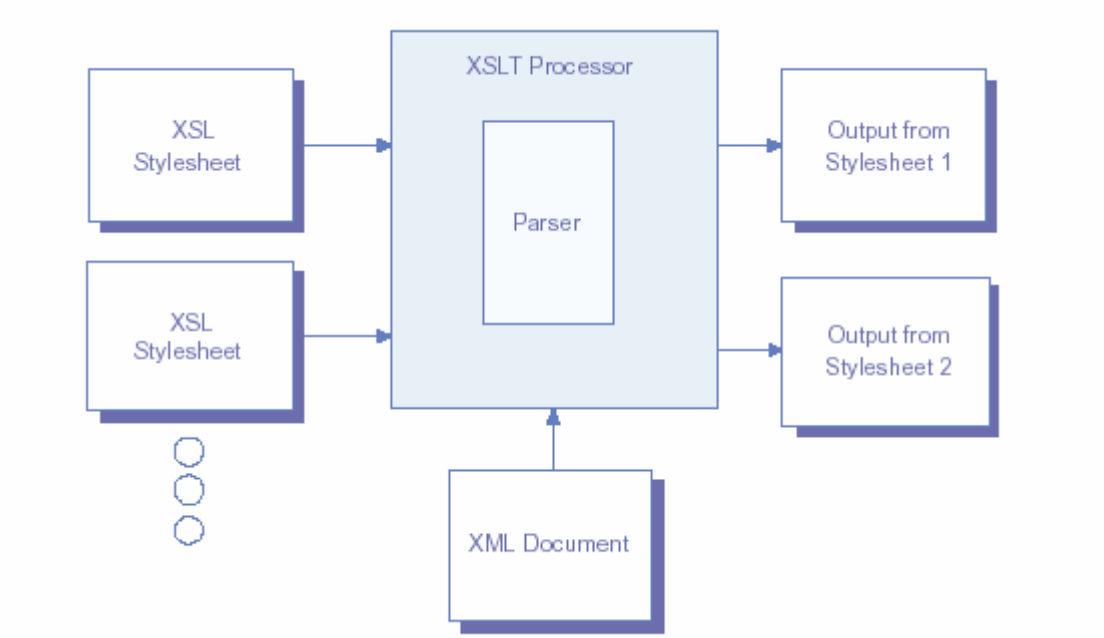
Look at the new tags!
A Human and a computer can now easily extract the publisher data.

XML parser



- An XML *parser* is a software component that can read and validate any XML document. A parser makes data contained in an XML data structure available to the application that needs to use it

XML stylesheet



- A key advantage of XML over other data formats is the ability to convert an XML document to another form
- An XSL stylesheet is a set of transformation instructions to converting a *source* XML document to a *target* document

Validation of XML Documents

- XML documents *must* be well-formed
- XML documents *may* be valid
  - Validation verifies that the structure and content of the document follows rules specified by grammar
- Types of grammars
  - Document Type Definition (DTD)
  - XML Schema (XSD)
  - Relax NG (RNG)

DTD

- Document Type Definition
  - Defined in the XML 1.0 specification
  - Allows user to create new document grammars
    - A subset borrowed from SGML
    - Uses non-XML syntax!
  - Document-centric
    - Focus on document structure
    - Lack of “normal” datatypes (e.g. int, float)
- DTD limitations
  - Simple document structures
  - Lack of “real” datatypes

Example DTD (1 of 6)

- Text declaration

```
01    <?xml version='1.0' encoding='ISO-8859-1'?>
02    <!ELEMENT order (item)+>
03    <!ELEMENT item (name,price)>
04    <!ATTLIST item code NMTOKEN #REQUIRED>
05    <!ELEMENT name (#PCDATA)>
06    <!ELEMENT price (#PCDATA)>
07    <!ATTLIST price currency NMTOKEN 'USD'>
```

Example DTD (2 of 6)

- Element declarations

```
01    <?xml version='1.0' encoding='ISO-8859-1'?>
02    <!ELEMENT order (item)+>
03    <!ELEMENT item (name,price)>
04    <!ATTLIST item code NMTOKEN #REQUIRED>
05    <!ELEMENT name (#PCDATA)>
06    <!ELEMENT price (#PCDATA)>
07    <!ATTLIST price currency NMTOKEN 'USD'>
```

Example DTD (3 of 6)

- Element content models

```
01    <?xml version='1.0' encoding='ISO-8859-1'?>
02    <!ELEMENT order (item)+>
03    <!ELEMENT item (name,price)>
04    <!ATTLIST item code NMTOKEN #REQUIRED>
05    <!ELEMENT name (#PCDATA)>
06    <!ELEMENT price (#PCDATA)>
07    <!ATTLIST price currency NMTOKEN 'USD'>
```

Example DTD (4 of 6)

- Attribute list declarations

```
01    <?xml version='1.0' encoding='ISO-8859-1'?>
02    <!ELEMENT order (item)+>
03    <!ELEMENT item (name,price)>
04    <!ATTLIST item code NMTOKEN #REQUIRED>
05    <!ELEMENT name (#PCDATA)>
06    <!ELEMENT price (#PCDATA)>
07    <!ATTLIST price currency NMTOKEN 'USD'>
```

Example DTD (5 of 6)

- Attribute value type

```
01    <?xml version='1.0' encoding='ISO-8859-1'?>
02    <!ELEMENT order (item)+>
03    <!ELEMENT item (name,price)>
04    <!ATTLIST item code NMTOKEN #REQUIRED>
05    <!ELEMENT name (#PCDATA)>
06    <!ELEMENT price (#PCDATA)>
07    <!ATTLIST price currency NMTOKEN 'USD'>
```

Example DTD (6 of 6)

- Attribute default value

```
01    <?xml version='1.0' encoding='ISO-8859-1'?>
02    <!ELEMENT order (item)+>
03    <!ELEMENT item (name,price)>
04    <!ATTLIST item code NMTOKEN #REQUIRED>
05    <!ELEMENT name (#PCDATA)>
06    <!ELEMENT price (#PCDATA)>
07    <!ATTLIST price currency NMTOKEN 'USD'>
```

XML Namespaces

- The problem
  - Documents use different vocabularies
  - Merging multiple documents together
- The solution
  - Namespace: a syntactic way to differentiate similar names in an XML document
  - Binding namespaces
    - Uses Uniform Resource Identifier (URI)
      - e.g. “http://example.com/NS”
    - Can bind to a named or “default” prefix

Namespace Binding

- Use “xmlns” attribute
  - Named prefix
    - e.g. <a:foo xmlns:a=‘http://example.com/NS’/>
  - Default prefix
    - e.g. <foo xmlns=‘http://example.com/NS’/>
- Element and attribute names are “qualified”
  - URI, local part (or “local name”) pair
    - e.g. { “http://example.com/NS” , “foo” }

Example Document (1 of 3)

- Namespace binding

```
01      <?xml version='1.0' encoding='UTF-8'?>
02      <order>
03        <item code='BK123'>
04          <name>Care and Feeding of Wombats</name>
05          <desc xmlns:html='http://www.w3.org/1999/xhtml'>
06            The <html:b>best</html:b> book ever written!
07          </desc>
08        </item>
09      </order>
```

Example Document (2 of 3)

- Namespace scope

```
01    <?xml version='1.0' encoding='UTF-8'?>
02    <order>
03      <item code='BK123'>
04        <name>Care and Feeding of Wombats</name>
05        <desc xmlns:html='http://www.w3.org/1999/xhtml'>
06          The <html:b>best</html:b> book ever written!
07        </desc>
08      </item>
09    </order>
```

Example Document (3 of 3)

- Bound elements

```
01      <?xml version='1.0' encoding='UTF-8'?>
02      <order>
03        <item code='BK123'>
04          <name>Care and Feeding of Wombats</name>
05          <desc xmlns:html='http://www.w3.org/1999/xhtml'>
06            The <html:b>best</html:b> book ever written!
07          </desc>
08        </item>
09      </order>
```

XML Schema (XSD)

- A grammar definition language
  - Like DTDs but better
    - Uses XML syntax
  - But there are many schemas; there's no way to relate schema
- Primary features
  - Datatypes
    - e.g. integer, float, date, etc…
  - More powerful content models
    - e.g. namespace-aware, type derivation, etc…

XML Schema Types

- Simple types
  - Basic datatypes
  - Can be used for attributes *and* element text
  - Extendable
- Complex types
  - Defines structure of elements
  - Extendable
- Types can be named or “anonymous”

Simple Types

- DTD datatypes
  - Strings, ID/IDREF, NMTOKEN, etc…
- Numbers
  - Integer, long, float, double, etc…
- Other
  - Binary (base64, hex)
  - QName, URI, date/time
  - etc…

Deriving Simple Types

- Apply facets
  - Specify enumerated values
  - Add restrictions to data
  - Restrict lexical space
    - Allowed length, pattern, etc…
  - Restrict value space
    - Minimum/maximum values, etc…
- Extend by list or union

A Simple Type Example (1 of 4)

- Integer with value (1234, 5678]

```
01      <xsd:simpleType name='MyInteger'>
02        <xsd:restriction base='xsd:integer'>
03          <xsd:minExclusive value='1234'/>
04          <xsd:maxInclusive value='5678'/>
05        </xsd:restriction>
06      </xsd:simpleType>
```

A Simple Type Example (2 of 4)

- Integer with value (1234, 5678]

```
01      <xsd:simpleType name='MyInteger'>
02        <xsd:restriction base='xsd:integer'>
03          <xsd:minExclusive value='1234'/>
04          <xsd:maxInclusive value='5678'/>
05        </xsd:restriction>
06      </xsd:simpleType>
```

A Simple Type Example (3 of 4)

- Integer with value (1234, 5678]

```
01      <xsd:simpleType name='MyInteger'>
02        <xsd:restriction base='xsd:integer'>
03          <xsd:minExclusive value='1234'/>
04          <xsd:maxInclusive value='5678'/>
05        </xsd:restriction>
06      </xsd:simpleType>
```

A Simple Type Example (4 of 4)

- Validating integer with value (1234, 5678]

```
01    <data xsi:type='MyInteger'></data>          INVALID
02    <data xsi:type='MyInteger'>Andy</data>      INVALID
03    <data xsi:type='MyInteger'>-32</data>       INVALID
04    <data xsi:type='MyInteger'>1233</data>      INVALID
05    <data xsi:type='MyInteger'>1234</data>      INVALID
06    <data xsi:type='MyInteger'>1235</data>
07    <data xsi:type='MyInteger'>5678</data>
08    <data xsi:type='MyInteger'>5679</data>      INVALID
```

Complex Types

- Element content models
  - Simple
  - Mixed
    - Unlike DTDs, elements in mixed content can be ordered
  - Sequences and choices
    - Can contain nested sequences and choices
  - All
    - All elements required but order is *not* important

XML-based Messaging

- The most fundamental underpinning of WS architecture is XML Messaging
- XML as the basis for the messaging protocol
- The current industry standard for XML messaging is SOAP
- IBM, Microsoft, Sun and others submitted SOAP to the W3C
- SOAP
  - Is a simple and lightweight XML-based mechanism for exchanging structured data beetween network applications

Limitations for Semantic Markup

- XML makes no commitment on:
  - Domain-specific ontological vocabulary
  - Ontological modeling primitives
  - Requires pre-arranged agreement on vocabulary ar ontology
- Only feasible for closed collaboration
  - agents in a small & stable community
  - pages on a small & stable intranet
- Not suited for sharing Web-resources

Metadata

- Metadata
- Type
- Model

Hierarchy

- Object level vs. Meta level
  - meta meta level
- Object language vs. Metalanguage
  - formal language (first order language, programming language)
  - natural language
- Data vs. Metadata
  - database
  - XML

Metadata

- Metadata is “Data” about “Data”.
- Metadata is descriptive information for resources to be described.
- Specifically, metadata is machine understandable information for the Web.
- metadata is a descriptor or an identifier
  - Person name, address, identifier
  - Documen ttitle, keyword, version
  - Image name, size, location
  - Sound/Music title, time, creator

Why Need Metadata

- To describe various data accessed through the Internet and hence to improve the accessibility of the web resources
- To enhance the interchangeability between the web resources
- To talk to different existing formats for descriptions of information

Metadata Type

- Four types of metadata, which can be used to describe an electronic document or resources along different dimensions
  - Content information
  - Managerial information
  - Referencing information
  - Carrier information

Content Information

- The information describes the object content.

  E.g., consider some web page: headlines, subjects, introductory paragraphs, together with images, movies, etc.

  Within an article: subtitles, section titles, keywords, review comments, etc.

  By browsing the descriptive information, the web readers can quickly figure out what the content is all about.

Managerial Information

- The data items used to describe the relevant to an electronic document, such as author, creator, creation date, etc.

  E.g., version: versioning is an important measure of the evolutionary process of an object, this attribute is a dynamic factor about the document. The managerial information is essential when we want to know some “publication” information about the objects of interest.

Referencing Information

- The links reference among the recourses, comes from the “hyperlinks” appearing in an electronic document.

  The environment information can be also called reference information, which means that there may be other objects or resources associated to the focused object and used for detailed descriptions of the object.

  E.g., a paper’s structure is presented with a set of links to its various chapters, which appear in other web resources. The referencing structure for an electronic document can be hierarchical, where an upper object has several links to its children object, and neighboring.

Carrier Information

- The physical attributes about an electronic document.

  E.g., fonts, color, size, etc.

  These information pieces become important when some semantic meanings are assigned to them.

  E.g., “bold face” of a text may mean that the text is emphasized.

  In email systems, people may hope to control the size of emails. Templates’ information (layout) of electronic documents is also included in the carrier information, because this can help to manage different formats for documents.

Metadata Models

- PICS (Platform for Internet Content Selection) for rating electronic documents
- DC (Dublin Core) for publication description
- XML for general description of documents
- RDF (Resource Description Framework) for the web information description
- WDM+ (Web Document Model) for electronic document management
- RDFS (RDF Schema) for the Web information semantics representation
- DAML+OIL (DARPA Agent Markup Language + Ontology Interface Layer) - a semantic markup language for Web resources
- OWL (Web Ontology Language) - a semantic markup language for publishing and sharing ontologies on the Web.

Platform for Internet Content Selection

- Platform for Internet Content Selection (PICS) was developed to provide users with the ability to select content based on labels provided by information providers or other sources.
  - rating system description (a sort of schema)
  - every PICS label points to a description
- PICS was designed as a first step toward generalized labels that would allow any party in the Web to make claims about the qualities of resources: endorsements, terms and conditions for use, etc.

  The Metadata Activity of W3C addresses the necessary work to complete the picture: structured labels, rules, integration with digital signatures.
- PICS label design was generalized to a model of information as directed labelled graphs (DLGs). This was known as the RDF model, and a serialization was defined in XML syntax. PICS rating systems were incorporated as special cases in the design of RDF Schemas.

Rating System for MPAA

- The Motion Picture Association of America (MPAA) and its international counterpart, the Motion Picture Association (MPA) serve as the voice and advocate of the American motion picture, home video and television industries, domestically through the MPAA and internationally through the MPA.

```
((PICS-version 1.1)
        (rating-system "http://MPAAscale.org/Ratings/Description/")
        (rating-service "http://MPAAscale.org/v1.0")
        (icon "icons/MPAAscale.gif")
        (name "The MPAA's Movie-rating Service")
        (description "A rating service based on the MPAA's movie-rating scale")
        (category
                (transmit-as "r")
                (name "Rating")
                (label (name "G") (value 0) (icon "icons/G.gif"))
                (label (name "PG") (value 1) (icon "icons/PG.gif"))
                (label (name "PG-13") (value 2) (icon "icons/PG-13.gif"))
                (label (name "R") (value 3) (icon "icons/R.gif"))
                (label (name "NC-17") (value 4) (icon "icons/NC-17.gif"))
        )
)
```

Rating System for RSAC

- The Recreational Software Advisory Council (RSAC)use their own (copyrighted) rating system, which contains four categories: Violence, Nudity, Sex, and Language. Each category is rated on a scale from 0 to 4, with a specific description for each value. Only values with names are permitted.

PICS Labelling

(PICS-1.1 "http://old.rsac.org/v1.0/" labels
on "1994.11.05T08:15-0500"
until "1995.12.31T23:59-0000"
for "http://www.gcf.org/stuff.html"
by "John Doe"
ratings (l 3 s 2 v 0))

- According to RSAC:
- l=language, s=sex, v=violence

Dublin Core

- DC Metadata in an HTML document

```
<html>
 <head>
  <title>Distributed Metadata</title>
  <meta name="description" content="This article addresses...">
  <meta name="subject" content="metadata, rdf, peer-to-peer">
  <meta name="creator" content="Dan Brickleyand RaelDornfest">
  <meta name="publisher" content="O'Reilly & Associates">
  <meta name="date" content="2000-10-29T00:34:00+00:00">
  <meta name="type" content="article">
  <meta name="language" content="en-us">
  <meta name="rights" content="Copyright 2000, O'Reilly & Associates, Inc.">
  …
 </head>
```

DC Elements in Categories

Content	Intellectual Property	Instantiation
Title	Creator	Date
Subject	Publisher	Type
Description	Contributor	Format
Source	Rights	Identifier
Language		
Relation		
Coverage		

DC 15 Elements

- **Title** The name of the object, given by the creator or the publisher.
- **Subject** The topic addressed by the object, including keywords for the resourceor the content.
- **Description** Description of the resource content.
- **Source** Objects, either print or electronic, from which this object is derived.
- **Language** Language of the intellectual content.
- **Relation** Relationship to other objects.
- **Coverage** The spatial locations and temporal duration characteristic of the object.
- **Creator** Responsible for the content (intellectual) of the resource/object
- **Publisher** The agent or agency responsible for making the object available.
- **Contributor** The person(s) primarily responsible for the intellectual contentof the object.
- **Rights** (An identifier of) A statement of right management of the resource.
- **Date** The date of publication.
- **Type** The genre of the object, such as novel, poem, or dictionary.
- **Format** The data representation of the object, such as PostScript file.
- **Identifier** String or number used to uniquely identify the object, such as URL or URN.

Web Document Model

- A huge pile of electronic documents (resources):
  - Documents: body information (content)
  - Descriptions: format information (metadata)
  - Controls/operations: signature, send/receive
- Modeling:
  - Body attributes (content structure): headings, images
  - Identifiers + descriptive attributes: URL, authors, creators, ratings
  - Controls/operations/communications: updates, move-aways
- Requirements:
  - “Super” - a more general description of the web information
  - Simple - easy to create and use
  - Expressive - able to describe the web resources and information of diversity

WDM(contd.)

- Principles (in accordance to RDF)
  - Identifying objects by attributes, related objects/neighbors, and relationships/links
- Model
  - Objects (resources), relationships (links, associations), attributes (properties)
- Usage
  - Identification, classification, filtering, reuse and blocking
- Dynamic features of e-documents:
  - Document versioning
  - Metadata updating
  - Referencing resources moved away

Ontology

- What're Ontologies
- What're They for

Motivation

- Vast information is available on the Web
- Growing need for
  - Finding relevant information *(Information Extraction)*
  - Creating new knowledge out of the available information *(Web Content Mining)*
  - Personalization of the web
  - Learning about customers or individual users *(Web Usage Mining)*

Ontology

Philosophy

- Theory of what exists in the world

Logic

- Theory of what exists in the domain

Computer Science

- Formal description of shared concepts in a domain
- A vocabulary for the domain knowledge (AI)

Web

- Formal description of shared concepts of information resources on the Web

Definition

- An ontology is an explicit specification of a conceptualization
- A conceptualization is an abstract, simplified view of the world that we want to represent
- If the specification medium is a formal language, the ontology defines a representational foundation

Ontology (cont.)

Typically, an ontology contains:

- Classes – the objects of a domain

  each class is characterized by properties shared by all elements in that class
- Relations – relations between classes or between the elements of the classes
- Hierarchies - organizes these classes in a subclass hierarchy

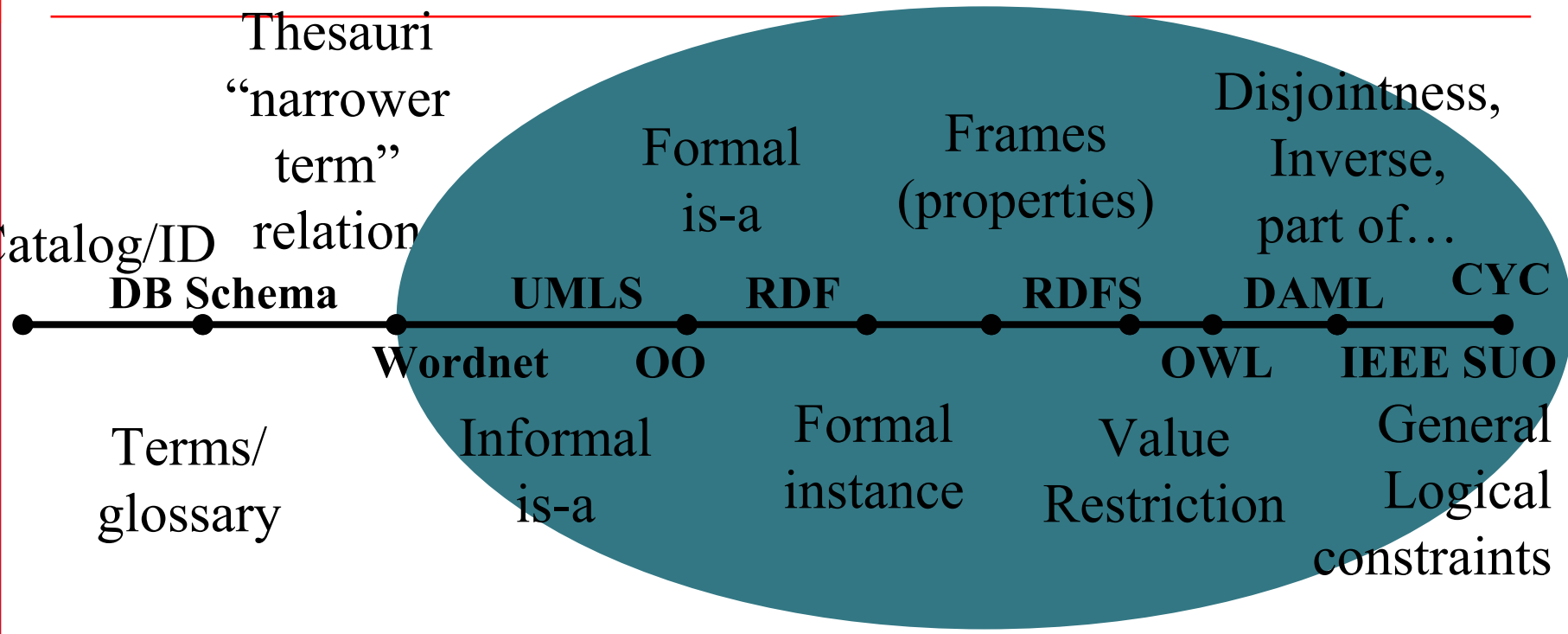
Ontology (cont.)

- The Knowledge-level: a level of description of the knowledge of an agent that is independent of internal format
  - *An agent "knows" if it acts like it does*
  - *A software agent "acts" by telling and asking*
- An agent commits (conforms) to an ontology if it "acts" consistently with the definitions
  - *Ontological Commitments are agreements to use the vocabulary in a coherent and consistent manner*
  - *Common ontology ≠common knowledge*

What Isn't an ontology

- A database or program
  - *because they share internal formats*
- a conceptualization
  - *because it isn't a specification -it's a vision*
- a table of contents
  - *but isn't a Taxonomy an Ontology?*
  - *only if it defines a set of concepts*

History

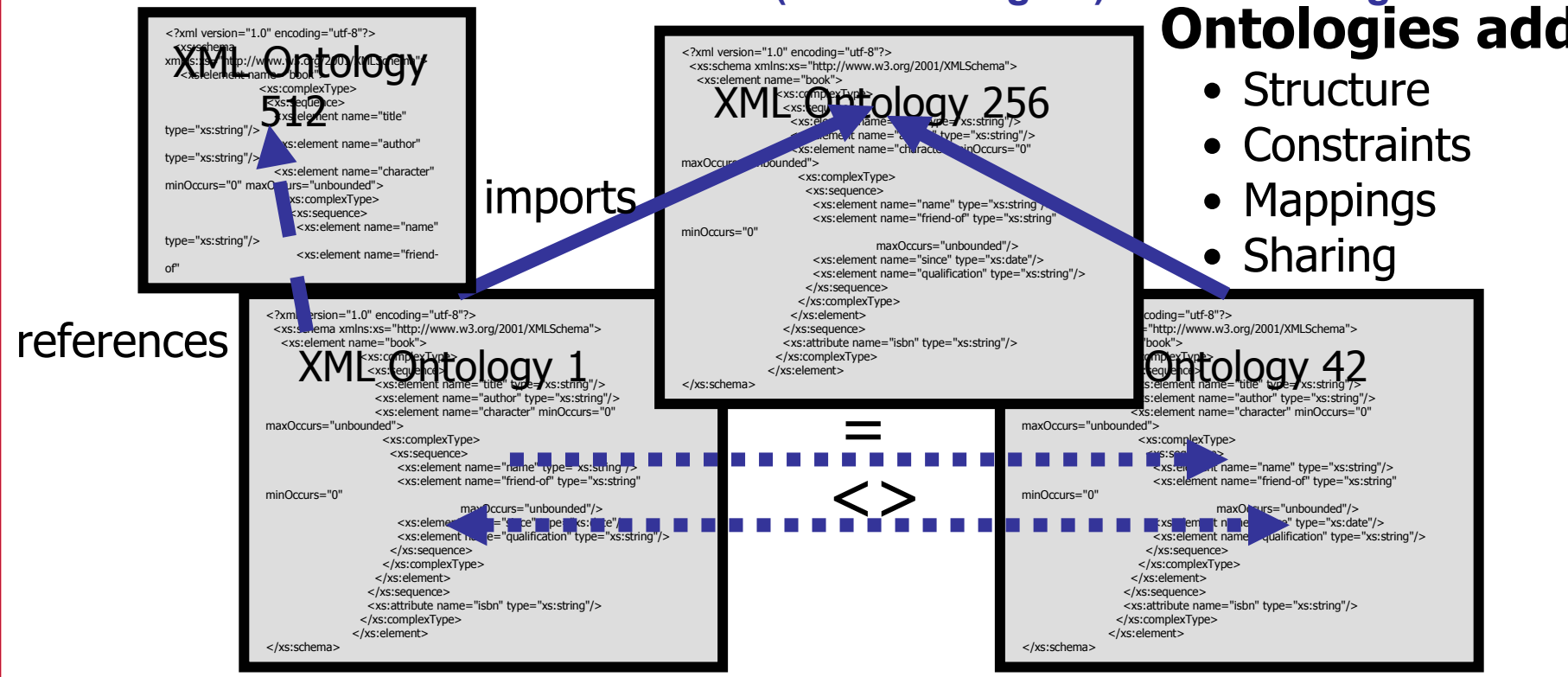


Simple **Expressive**

Taxonomies **Ontologies**

An Ontological Level Is Needed

- We need a way to define ontologies in XML
  - **So we can relate them**
  - **So machines can understand (to some degree) their meaning**



Why Create Ontologies

- to enable data exchange among programs
- to simplify unification (or translation) of disparate representations
- to employ knowledge-based services
- to embody the representation of a theory
- as a reference to guide new formalizations
- to facilitate communication among people

Dublin Core - A Simple Ontology

- Developed by an OCLC sponsored workshop in Dublin ~95 as a standard for metadata for digital library resources on web
  - Consists of 15 core attributes
  - http://dublincore.org/
- Neutral on how DC should be represented
- HTML found to be inadequate for representing complexities of structured use of DC
- Available as an RDF schema.

15 DC elements

Content elements

- Coverage
- Description
- Relation
- Source
- Subject
- Title
- Type

Intellectual Property

- Contributor
- Creator
- Publisher
- Right

Instantiation

- Date
- Format
- Identifier
- Language

CYC - A Complex Ontology

- Cyc is a large, general purpose ontology with associated reasoning tools developed over the past 20 years by MCC and now Cycorp
  - Cyc KB has > 100k terms
  - Terms are axiomatized by > 1M handcrafted assertions
  - Cyc inference engine has > 500 heuristic level modules
- Goal is to encode knowledge for "common sense reasoning" needed by applications (e.g., NLP)
- Available free in limited form from http://opencyc.org/