Metadata and the Semantic Web

Marlon Pierce
Community Grids Lab
Indiana University

Introduction

Overview things and provide some motivating examples.

Some Definitions from the W3C

"The **Semantic Web** is the representation of **data** on the **World Wide Web**. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners. It is based on the **Resource Description Framework** (**RDF**), which integrates a variety of applications using **XML** for syntax and **URIs** for naming."

Introduction: What's in data.out?

- Metadata is information about other data-->resources.
  - Typically it is a collection of property-value pairs
    - Property names established by convention
  - What is my data?
  - Where is my data?
  - When was this data created?
  - How was it created?
  - Who generated this garbage?
- The Semantic Web attempts to define a metadata information model for the Internet to aid in information retrieval and aggregation.
  - Provides general languages for describing any metadata
  - Advanced capabilities intended to enable knowledge representation and limited machine reasoning.
- Coupling of the Semantic Web with Web Service and Grid technologies is referred to as the Semantic Grid.

Seminar Goal

- Demonstrate the wide applicability of XML metadata representation to
  - DOD Scientists: describing data provenance, computing resources
  - Information managers: digital libraries
- Hear from participants
  - Are you using metadata now?
  - Do you see the need for this in your research?

Where Are the Agents?

- The statements says nothing about intelligent agents scouring the web.
- The SW data descriptions are intended to provide information encoding for software applications.
- Well known Scientific American article by Tim Berners-Lee, James Hendler, and Ora Lassila
  - http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21
  - Agents schedule doctor's visit and itinerary for several people.
- More prosaically, SW can potentially enable more sophisticated searches.
  - Linked searches: return to me all of the citations of papers by M. Pierce that do not have M. Pierce in the author list.
- Potentially enable lightweight data federation:
  - Generic descriptions that can span databases, HTML web pages, web accessible documents.

Structure of the Talk

- Metadata: What Is It?
- Semantic Web Ideas and Languages
  - RDF, RDFS, DAML-OIL, and OWL
- Information Modeling with RDFS
  - An ontology for a semantic earthquake grid.
- Where is it?
  - Global issues
  - Some useful downloads and tools.

Semantic Web Vision

- Well known Scientific American article by Tim Berners-Lee, James Hendler, and Ora Lassila
  - http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21
- Example: making a doctor’s appointment

Preview Slide #1: RDF Structure

- RDF has a subject/verb/object structure.
  - “Semantic”
- It is represented in various ways.
  - XML
  - Graphs
  - Triples
- In RDF, everything is labeled with a URI.
  - Structured names
  - URLs are special cases.

```
<rdf:RDF>
 <rdf:Description
     about=‘Presentation’>
     <dc:creator
             rdf:resource=‘Marlon’/>
 </rdf:Description>
</rdf:RDF>
```

The pseudo-RDF sentence says “Presentation has creator Marlon”.

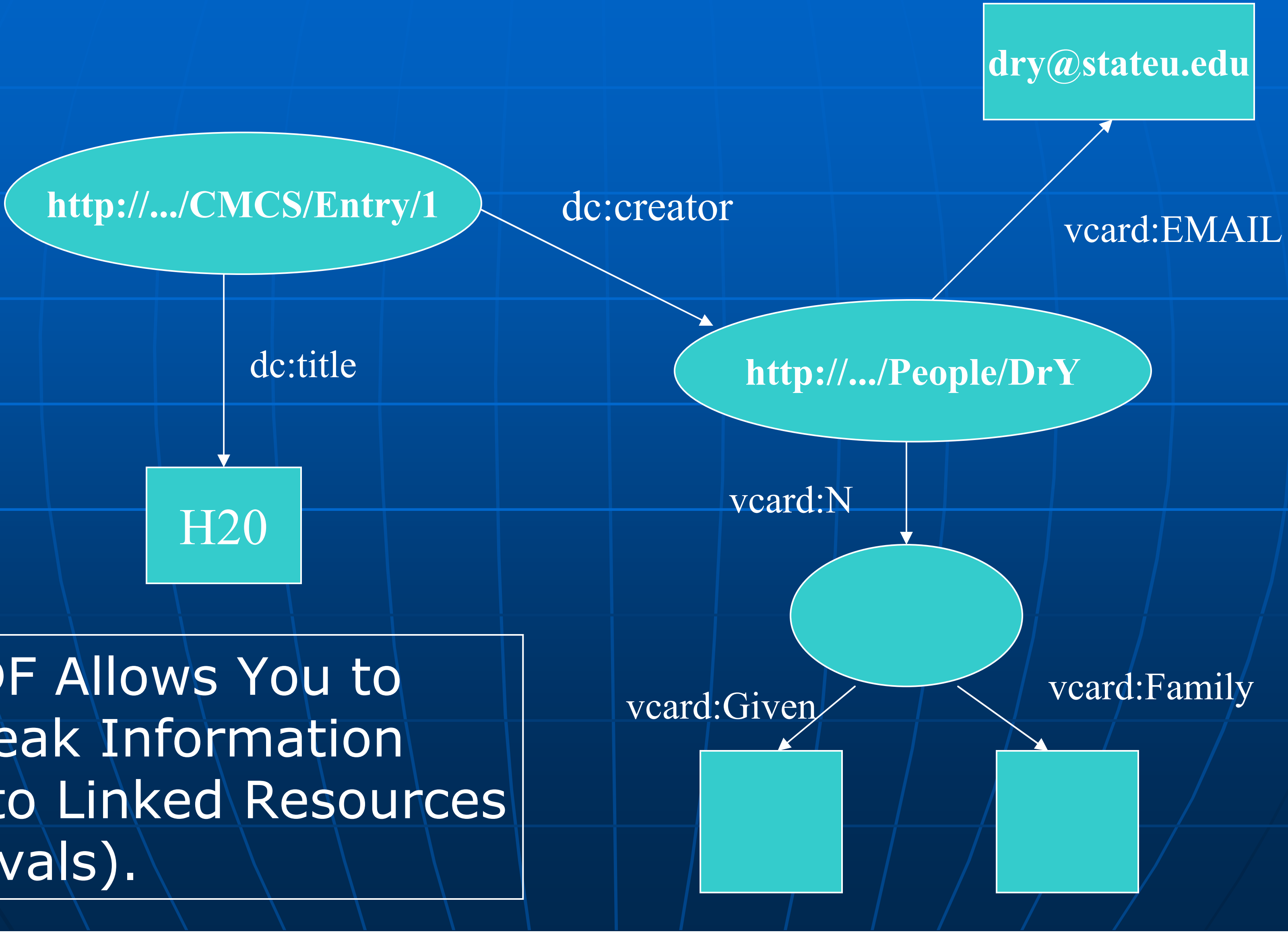
Preview Slide #2: Why Is XML Insufficient?

- RDF can be written in XML.
- But RDF is more:
  - XML defines syntax rules.

    ```
    <Parent name=Charles>
      <Child>Erasmus</Child>
    </Parent>
    ```

  - Relations between tags (meaning) is supplied by you.
    - "You" may be a large group, like the OGC.
  - The above XML nugget has many possible alternatives with the same "meaning".
- RDF attempts to encode semantic meaning.

Preview Slide #3: RDF Graphs



RDF Allows You to Break Information Into Linked Resources (Ovals).

Preview Slide #3: RDFS, DAML-OIL and OWL

- As we will see, RDF is extremely simple.
  - Envelope for holding metadata.
  - Properties link resources (subjects and objects).
  - Everything is named with a URI.
- RDF Schema is a way to build RDF dialects
  - Defines conventions for creating properties.
- DAML-OIL and OWL are specific languages built from RDFS.
  - Define conventional properties for description logics (Inverse, exclusion, union, etc relations)
  - You can use these to build up ontologies for your field.

Scientific Metadata

Define metadata and describe its use in physical and computer science.

What is Metadata?

- Common definition: data about data
- "Traditional" Examples
  - Prescriptions of database structure and contents.
  - File names and permissions in a file system.
  - HDF5 metadata: describes scientific/numerical data set characteristics such as array sizes, data formats, etc.
- Metadata may be queried to learn the characteristics of the data it describes.
- Traditional metadata systems are functionally tightly coupled to the data they describe.
  - Prescriptive, needed to interact directly with data.

Descriptive Metadata and the Web

- Traditional metadata concepts must be extended as systems become more distributed, information becomes broader
  - Tight functional integration not as important
  - Metadata used for information, becomes descriptive.
  - Metadata may need to describe resources, not just data.
- Everything is a resource
  - People, computers, software, conference presentations, conferences, activities, projects.
- We’ll next look at several examples that use metadata, featuring
  - Dublin Core: digital libraries
  - CMCS: chemistry

The Dublin Core: Metadata for Digital Libraries

- The Dublin Core is a set of simple name/value properties that can describe online resources.
    - Usually Web content but generally usable (CMCS)
    - Intended to help classify and search online resources.
- DC elements may be either embedded in the data or in a separate repository.
- Initial set defined by 1995 Dublin, Ohio meeting.

Thought Experiment: Construct Your Own Metadata Set

- Describe yourself: your occupation, your interests, your place of residence, your parents, spouse, children,….
- Take each sentence:
  - The verbs become properties
  - The verbs' objects are property values.
- Metadata is just a collection of these name/value pairs.
- For particular fields (like publishing), we can define a conventional set of property names.

The Dublin Core: Metadata for Digital Libraries

- The Dublin Core is a set of simple name/value properties that can describe online resources.
  - Usually Web content but generally usable (CMCS)
  - Intended to help classify and search online library resources.
  - Digital library card catalog.
- DC elements may be either embedded in the data or in a separate repository.
- Initial set defined by 1995 Dublin, Ohio meeting.

Dublin Core Elements

- Content elements:
  - Subject, title, description, type, relation, source, coverage.
- Intellectual property elements:
  - Contributor, creator, publisher, rights
- Instantiation elements:
  - Date, format, identifier, language
- In RDF, these are called properties.

Encoding the Dublin Core

- DC elements are independent of the encoding syntax.
- Rules exist to map the DC into
  - HTML
  - RDF/XML
- We provide more detailed info on RDF/XML encoding in this seminar.

Sample RDF/HTML

```
 <head>
<title>Expressing Dublin Core in HTML/XHTML
   meta and link elements</title>
   <meta name="DC.title" content="Expressing
   Dublin Core in HTML/XHTML meta and link
   elements" />
   <meta name="DC.creator" content="Andy
   Powell, UKOLN, University of Bath" />
   <meta name="DC.type" content="Text" />
</head>
```

Where Do I Put the Dublin Core Metadata?

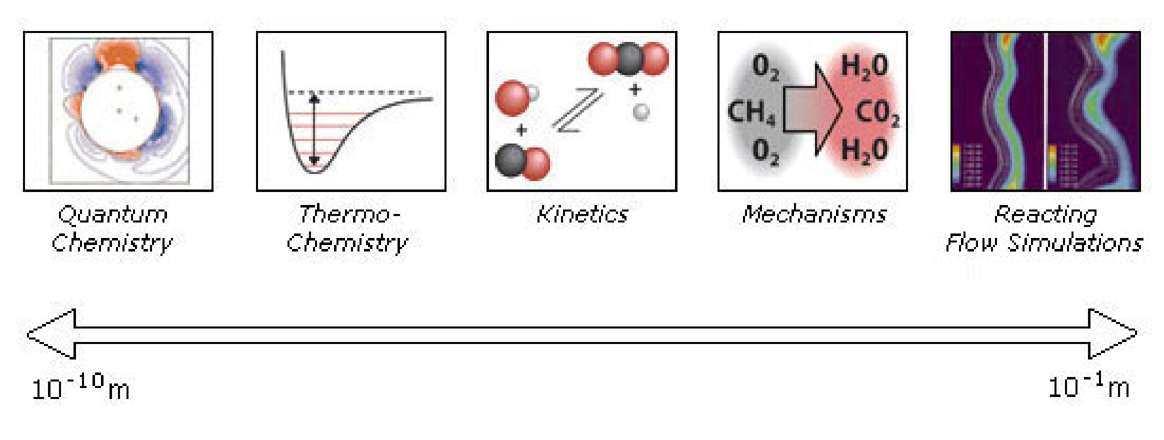
- Dublin core elements may be placed directly in HTML pages.
  - Still need DC-aware crawlers or applications to find and use them.
- Or you may have a large database on DC entries that are used by DC-aware applications.
  - We’ll examine a WebDAV-based scheme for chemistry in a second.

Dublin Core Element Refinements

- Many of these, and extensible
- See http://dublincore.org/documents/dcmi-terms/ for the comprehensive list of elements and refinements
- Examples:
  - isVersionOf, hasVersion, isReplacedBy, references, isReferencedBy.

Collaboratory for Multiscale Chemical Science (CMCS)

- SciDAC project involving several DOE labs
  - See http://cmcs.ca.sandia.gov/index.php.
- Project scope is to build Web infrastructure (portals, services, distributed data) to enable multiscale coupling of chemical applications

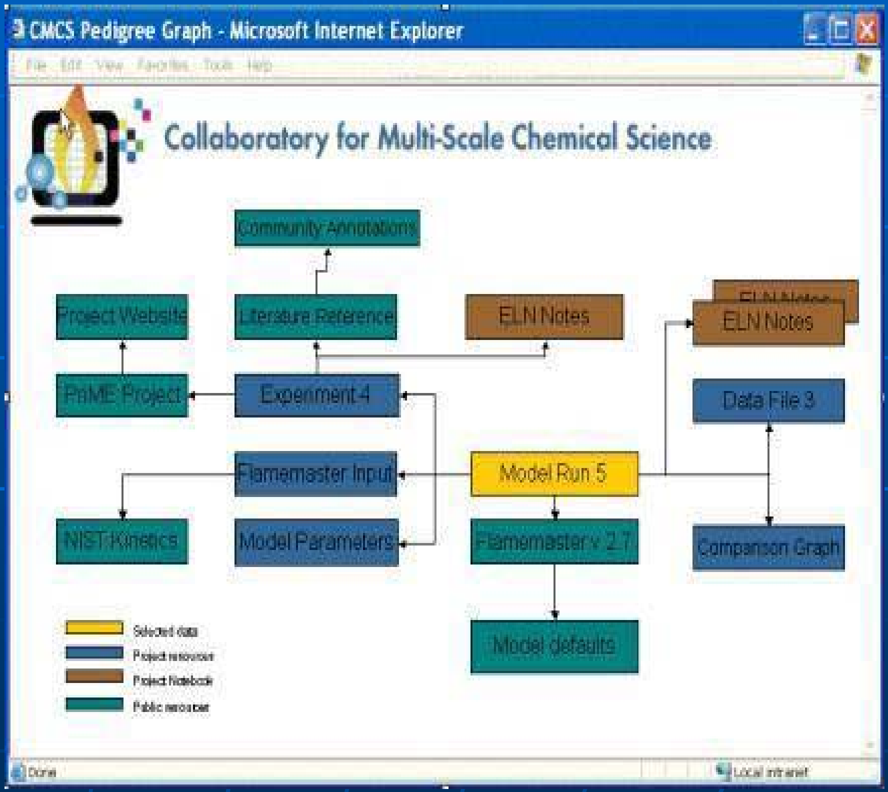


CMCS Is Data Driven Grid

- Core of the CMCS project is to exchange chemical data and information between different scales in a well defined, consistent, validated manner.
- Journal publication of chemical data is too slow.
- Need to support distributed online chemical data repositories.
- Need an application layer between the user and the data.
  - Simplify access through portals and intelligent search tools.
  - Control read/write access to data

CMCS Data Problems

- Users need to intelligently search repositories for data.
  - Characterize it with metadata
- Many data values are derived from long calculation chains.
  - Bad data can propagate, corrupt many dependent values.
- Experimental values are also sometimes questionable.
- Always the problem of incorrect data entry, errata.

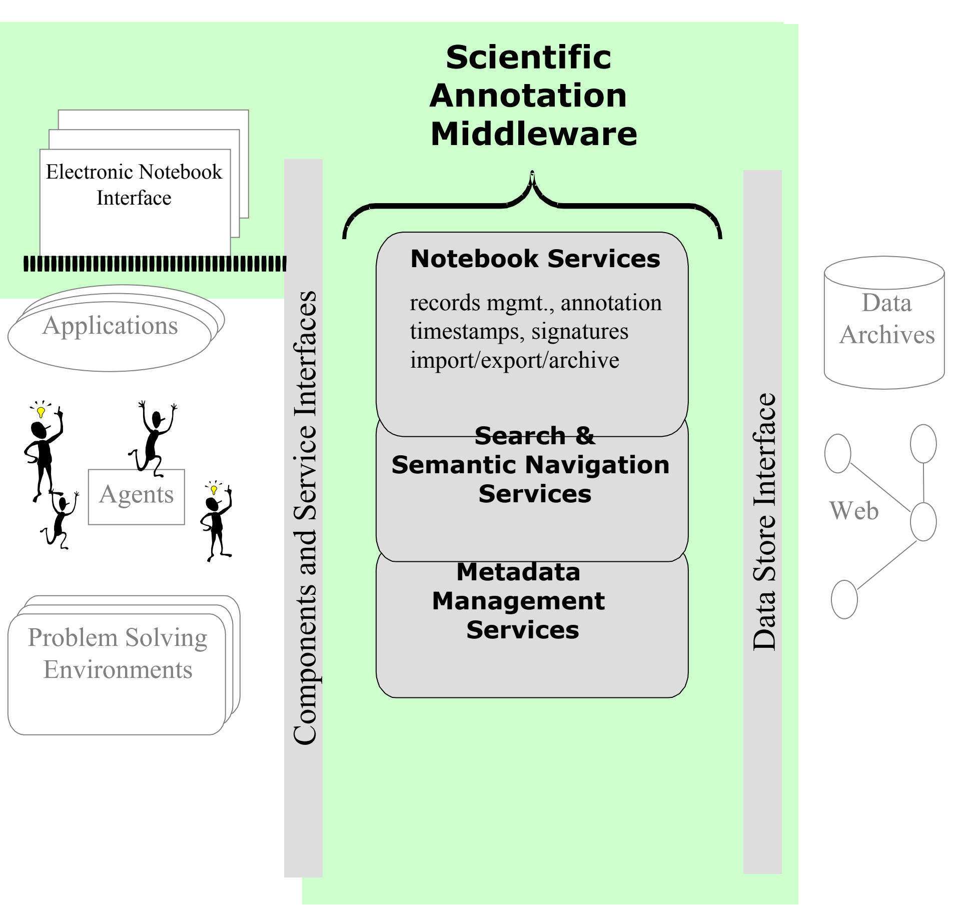


Solution: Annotation Metadata and Data Pedigree

- CMCS provides subject area metadata tags to identify data
  - Species name, Chemical Abstracts Service number, formula, common name, vibrational frequency, molecular geometry, absolute energy, entropy, specific heat, heat capacity, free energy differences, etc.
- Data Pedigree also must be recorded.
  - Where was it published/described?
  - Who measured or calculated the values?
    - Intellectual property
  - How were the values obtained?
  - What other values does it depend upon?
- Also provides community annotation capabilities
  - Is this value suspicious? Why?
    - Monte Carlo and other techniques exist to automate this.
  - Has the data been officially blessed? By whom?
    - Curation

Scientific Annotation Middleware (SAM) Approach

- General purpose metadata system.
- Based on WebDAV
  - Standard distributed authoring tool.
  - See next slide
- Uses extended Dublin Core elements to describe data.
  - Title, creator, subject, description, publisher, date, type, format, source,isResplaceBy, replaces, hasVersion isReferencedBy, hasReferences.
- SAM available from http://collaboratory.esml.pnl.gov/docs/collab/sam

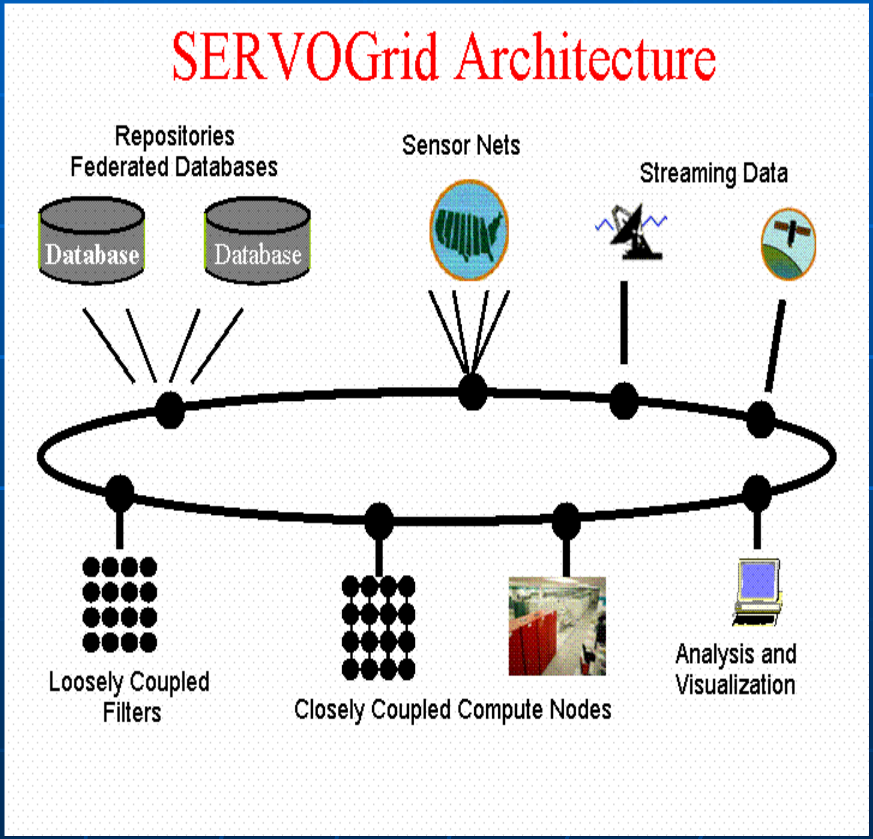

Scientific Annotation Middleware
Electronic Notebook Interface
Applications
Agents
Problem Solving Environments
Components and Service Interfaces
Notebook Services
records mgmt., annotation
timestamps, signatures
import/export/archive
Search & Semantic Navigation Services
Metadata Management Services
Data Store Interface
Data Archives
Web

Aside #0: What is WebDAV?

- IETF standard extension to HTTP for Web-based distributed authoring and version control.
  - Operations include put, get, rename, move, copy
  - Files are described with queryable metadata
    - Name/value pairs
    - Who is the author? What is the last revision?
  - Allows you to assign group controls
  - See many links at http://www.webdav.org/
- Web Service before its time.
- Documents for this seminar available from a WebDAV server (Slide).
- Many commercial implementations
  - MS Web folders are just DAV clients.

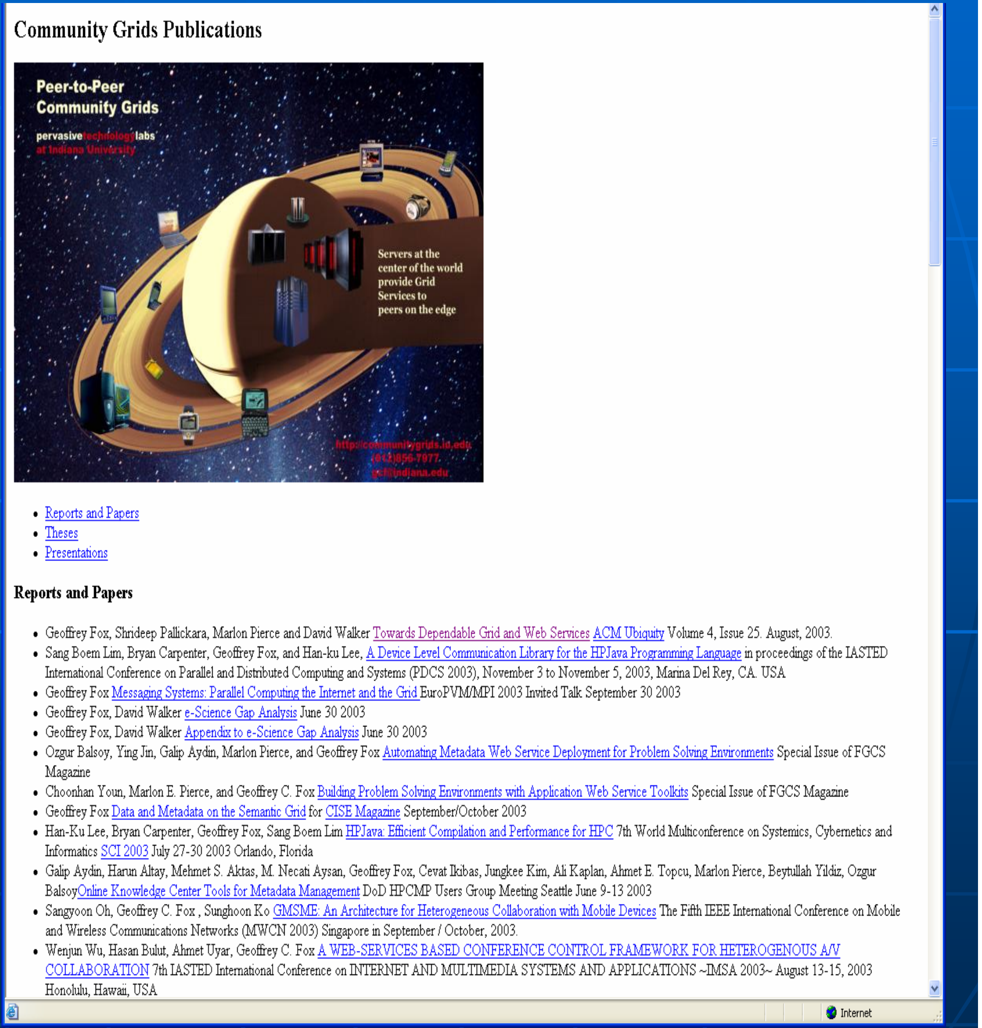
SERVOGrid

- CMCS issues are not unique to chemistry.
- SERVOGrid is a NASA project to integrate historical, measured, and calculated earthquake data (GPS, Seismicity, Faults) with simulation codes.
- Using GML extensions as common data format.
- "Those 1935 measurements aren't so good…"



Digital Libraries

- The CGL publication page is our simple "digital library".
  - http://grids.ucs.indiana.edu/ptliupages/publications/
- Raw material for testing tools and applications



Community Grids Publications

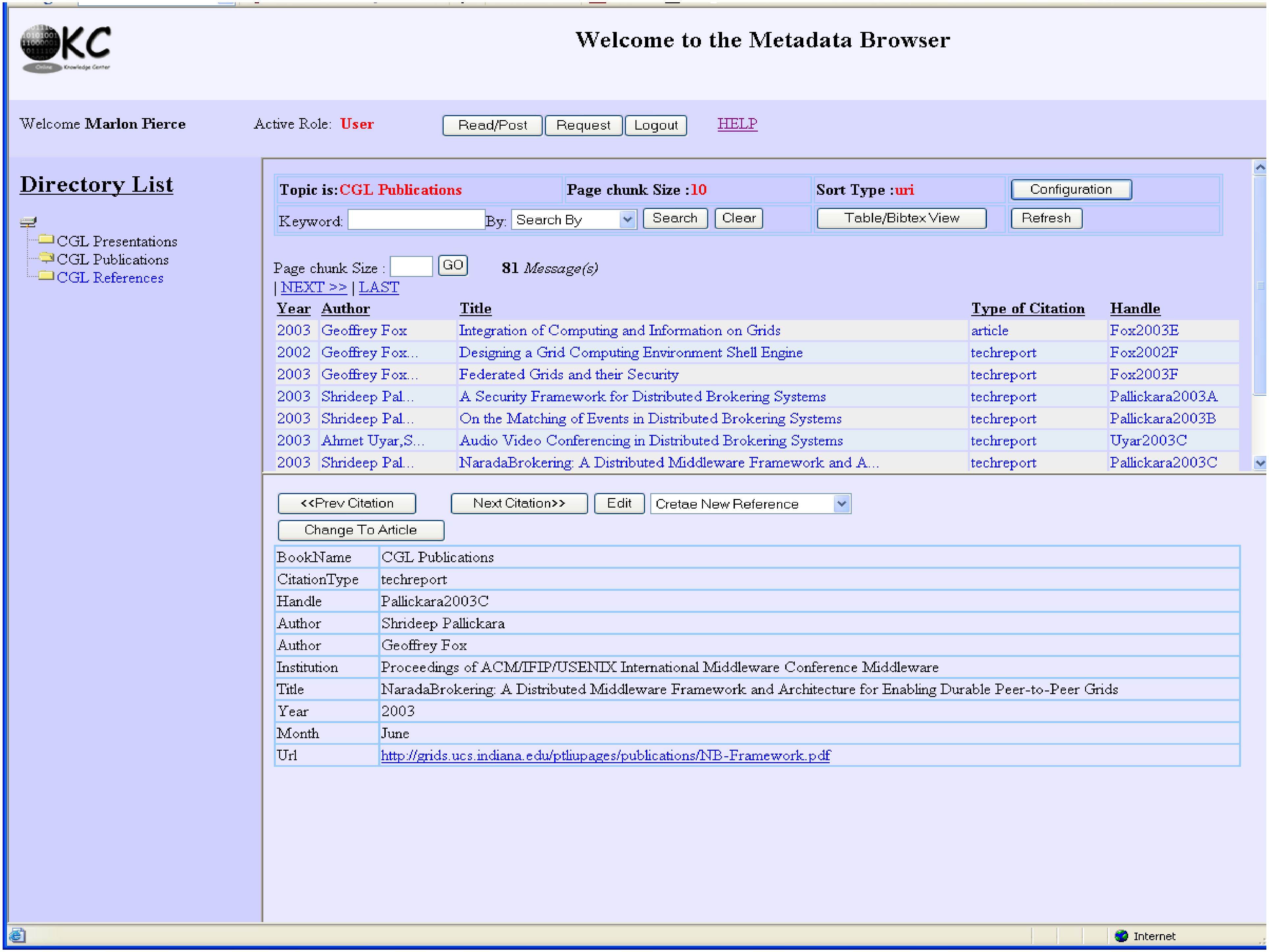
- Reports and Papers
- Theses
- Presentations

Reports and Papers

- Geoffrey Fox, Shrideep Pallickara, Marlon Pierce and David Walker Towards Dependable Grid and Web Services ACM Ubiquity Volume 4, Issue 25. August, 2003.
- Sang Boem Lim, Bryan Carpenter, Geoffrey Fox, and Han-ku Lee, A Device Level Communication Library for the HPJava Programming Language in proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2003), November 3 to November 5, 2003, Marina Del Rey, CA. USA
- Geoffrey Fox Messaging Systems: Parallel Computing the Internet and the Grid EuroPVM/MPI 2003 Invited Talk September 30 2003
- Geoffrey Fox, David Walker e-Science Gap Analysis June 30 2003
- Geoffrey Fox, David Walker Appendix to e-Science Gap Analysis June 30 2003
- Ozgur Balsoy, Ying Jin, Galip Aydin, Marlon Pierce, and Geoffrey Fox Automating Metadata Web Service Deployment for Problem Solving Environments Special Issue of FGCS Magazine
- Choonhan Youn, Marlon E. Pierce, and Geoffrey C. Fox Building Problem Solving Environments with Application Web Service Toolkits Special Issue of FGCS Magazine
- Geoffrey Fox Data and Metadata on the Semantic Grid for CISE Magazine September/October 2003
- Han-Ku Lee, Bryan Carpenter, Geoffrey Fox, Sang Boem Lim HPJava: Efficient Compilation and Performance for HPC 7th World Multiconference on Systemics, Cybernetics and Informatics SCI 2003 July 27-30 2003 Orlando, Florida
- Galip Aydin, Harun Altay, Mehmet S. Aktas, M. Necati Aysan, Geoffrey Fox, Cevat Ikibas, Jungkee Kim, Ali Kaplan, Ahmet E. Topcu, Marlon Pierce, Beytullah Yildiz, Ozgur BalsoyOnline Knowledge Center Tools for Metadata Management DoD HPCMP Users Group Meeting Seattle June 9-13 2003
- Sangyoon Oh, Geoffrey C. Fox , Sunghoon Ko GMSME: An Architecture for Heterogeneous Collaboration with Mobile Devices The Fifth IEEE International Conference on Mobile and Wireless Communications Networks (MWCN 2003) Singapore in September / October, 2003.
- Wenjun Wu, Hasan Bulut, Ahmet Uyar, Geoffrey C. Fox A WEB-SERVICES BASED CONFERENCE CONTROL FRAMEWORK FOR HETEROGENOUS A/V COLLABORATION 7th IASTED International Conference on INTERNET AND MULTIMEDIA SYSTEMS AND APPLICATIONS ~IMSA 2003~ August 13-15, 2003 Honolulu, Hawaii, USA

Lab Publications Also Maintained in XML Nuggets Browser

- Developed as part of the OKC work.
- Uses XML schemas based on bibtex for data models.
- Provides posting, browsing, searching, and editing features.
- Basic problem: Need a better way to link nuggets
  - Currently each application is based around a single schema or set of related schemas
  - Authors, publications, projects, glossary terms (key words) should all be navigable through a single interface.



Welcome to the Metadata Browser

Welcome **Marlon Pierce**   Active Role: **User**

Directory List

- CGL Presentations
- CGL Publications
- CGL References

Topic is:CGL Publications | **Page chunk Size :10** | **Sort Type :uri**

Keyword: By: Search By

Page chunk Size : **81** *Message(s)*

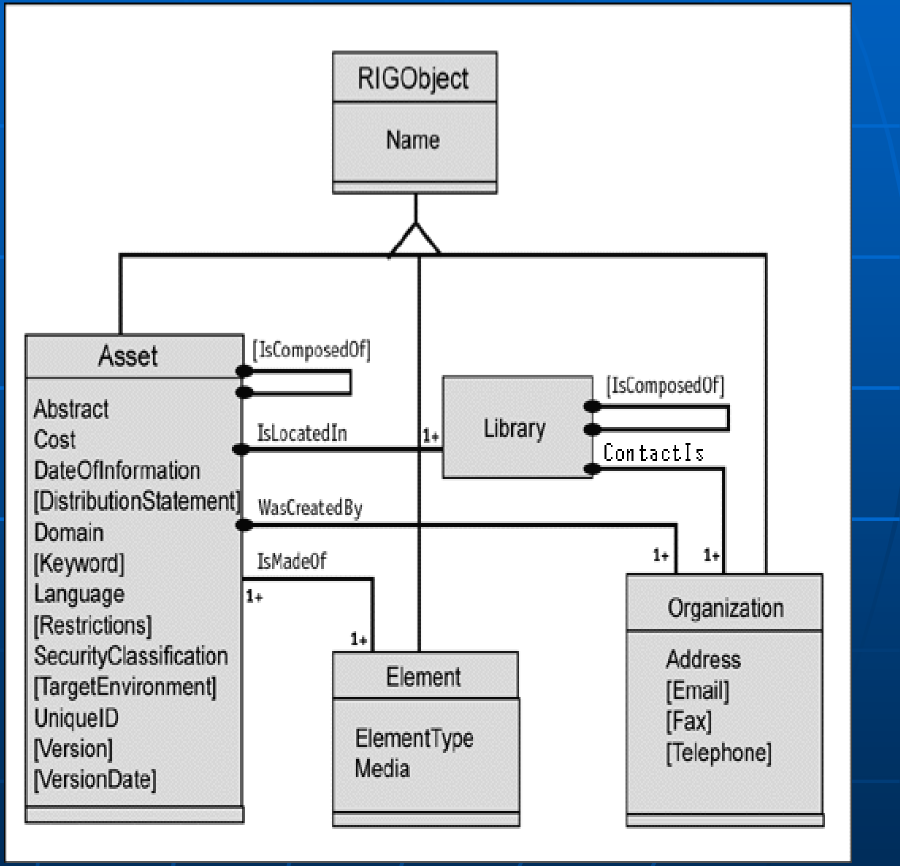
| NEXT >> | LAST

Year	Author	Title	Type of Citation	Handle
2003	Geoffrey Fox	Integration of Computing and Information on Grids	article	Fox2003E
2002	Geoffrey Fox...	Designing a Grid Computing Environment Shell Engine	techreport	Fox2002F
2003	Geoffrey Fox...	Federated Grids and their Security	techreport	Fox2003F
2003	Shrideep Pal...	A Security Framework for Distributed Brokering Systems	techreport	Pallickara2003A
2003	Shrideep Pal...	On the Matching of Events in Distributed Brokering Systems	techreport	Pallickara2003B
2003	Ahmet Uyar,S...	Audio Video Conferencing in Distributed Brokering Systems	techreport	Uyar2003C
2003	Shrideep Pal...	NaradaBrokering: A Distributed Middleware Framework and A...	techreport	Pallickara2003C

BookName	CGL Publications
CitationType	techreport
Handle	Pallickara2003C
Author	Shrideep Pallickara
Author	Geoffrey Fox
Institution	Proceedings of ACM/IFIP/USENIX International Middleware Conference Middleware
Title	NaradaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids
Year	2003
Month	June
Url	http://grids.ucs.indiana.edu/ptliupages/publications/NB-Framework.pdf

RIB: Software Metadata

- UTK’s RIB is designed manage software metadata.
- Assets are key class
  - Information about software specifications, documentation, source code, etc.
- Libraries are comprised of assets
- Organizations own libraries and assets.
- BIDM expressed in XML is current RIB versions.
- BIDM extensions express
  - Asset certification: curation
  - Intellectual property: pedigree



Universal Description Discovery and Integration (UDDI)

- General purpose online, integrated information repository.
  - Uses XML data models
  - Accessed through Web services (WSDL, SOAP).
- Often used as an information repository for Web services
  - Links to WSDL and service location URLs.
- Actually UDDI is a general purpose online business information system.
  - Can intended to store business information and classifications, contact information, etc.

Application and Computer Metadata

- Developed by IU to support Gateway project.
- Describes applications (codes), hosts, queuing systems.
  - Coupled XML schemas
- Stores information such as
  - Input and output file locations, machines used, etc.
- Coupled with Web services to run codes, generate batch scripts, archive portal sessions.

Grid Portal Information Repository (GPIR)

- TACC's GPIR is an integrated information system
  - XML data models (more below)
  - Web services for ingesting, querying data
  - Portlets for displaying, interacting with info
- Several independent XML schemas for describing computing resources
  - Static info: machine characteristics (OS, number of processors, memory) and MOTDs.
  - Dynamic info: loads, status of hosts, status of jobs on hosts, nodes on hosts.

Metadata Trends and Lessons

- Online data repositories becoming increasingly important, so need pedigree, curation, and annotation.
- XML is the method of choice for describing metadata.
  - "Human understandable"
  - OS and application independent.
  - Provides syntax rules but does not really encode meaning
- But there is no generic way to describe metadata.
  - How can we resolve differences in Application Metadata and GPIR, for example?
  - This should be possible, since metadata ultimately boils down to structured name/value pairs.
- The Semantic Web tools seek to solve these problems.

XML Primer

General characteristics of XML

Basic XML

- XML consists of human readable tags
- Schemas define rules for a particular dialect.
- XML Schema is the root, defines the rules for making other XML schemas.
- Tree structure: tags must be closed in reverse order that they are opened.
- Tags can be modified by attributes
  - name, minOccurs
- Tags enclose either strings or structured XML

```
<complexType name="FaultType">
   <sequence>
     <element name="FaultName"
                    type="xsd:string" />
     <element name="MapView/>
     <element name="CartView“/>
     <element name="MaterialProps"
                 minOccurs="0" />
     <choice>
        <element name="Slip" />
        <element name="Rate" />
     </choice>
  </sequence>
 </complexType>
```

Namespaces and URIs

- XML documents can be composed of several different schemas.
- Namespaces are used to identify the source schema for a particular tag.
  - Resolves name conflicts—”full path”
- Values of namespaces are URIs.
  - URI are just structured names.
    - May point to something not electronically retrievable
  - URLs are special cases.

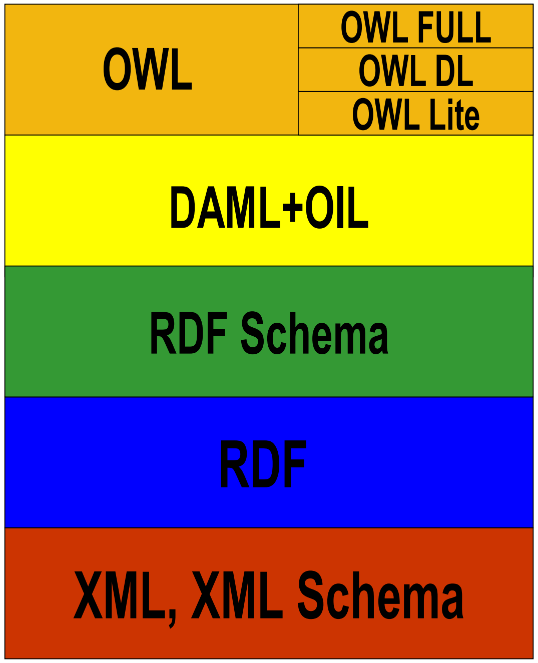
```
<xsd:schema
   xmlns:xsd="http://www.w
   3.org/2001/XMLSchema"
   xmlns:gem="http://comm
   grids.indiana.edu/GCWS/S
   chema/GEMCodes/Faults”
   >
  <xsd:annotation>
   …
 </xsd:annotation>
 <gem:fault>
   …
 </gem:fault>
</xsd:schema>
```

Resource Description Framework

Overview of RDF basic ideas and XML encoding.

Building Semantic Markup Languages

- XML essentially defines syntax rules for markup languages.
  - "Human readable" means humans provide meaning
- We also would like some limited ability to encode meaning directly within markup languages.
- The semantic markup languages attempt to do that, with increasing sophistication.
- Stack indicates direct dependencies: DAML is defined in terms of RDF, RDFS.



OWL | OWL FULL
OWL DL
OWL Lite

DAML+OIL

RDF Schema

RDF

XML, XML Schema

Resource Description Framework (RDF)

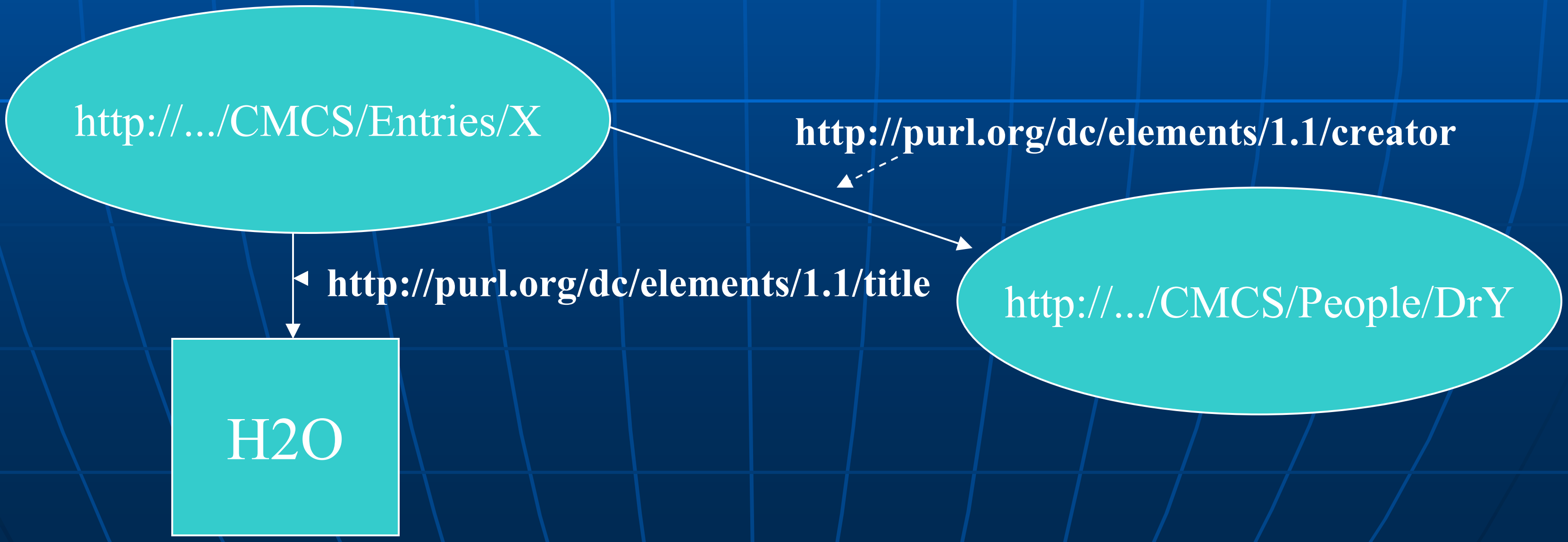
- RDF is the simplest of the semantic languages.
- Basic Idea #1: Triples
  - RDF is based on a subject-verb-object statement structure.
  - RDF subjects are called classes
  - Verbs are called properties.
- Basic Idea #2: Everything is a resource that is named with a URI
  - RDF nouns, verbs, and objects are all labeled with URIs
  - Recall that a URI is just a name for a resource.
  - It may be a URL, but not necessarily.
  - A URI can name anything that can be described
    - Web pages, creators of web pages, organizations that the creator works for,….

What Does This Have to Do with Grid Computing?

- RDF resources aren’t just web pages
  - Can be computer codes, simulation and experimental data, hardware, research groups, algorithms, ….
- Recall from the CMCS chemistry example that they needed to describe the provenance, annotation, and curation of chemistry data.
  - Compound X’s properties were calculated by Dr. Y.
- CMCS maps all of their metadata to the Dublin Core.
- The Dublin Core is encoded quite nicely as RDF.

RDF Graph Model

- RDF is defined by a graph model.
- Resources are denoted by ovals.
- Lines (arcs) indicate properties.
- Squares indicate string literals (no URI).
- Resources and properties are labeled by a URI.


http://.../CMCS/Entries/X
http://purl.org/dc/elements/1.1/creator
http://purl.org/dc/elements/1.1/title
http://.../CMCS/People/DrY
H2O

Encoding RDF in XML

- The graph represents two statements.
  - Entry X has a creator, Dr. Y.
  - Entry X has a title, H2O.
- In RDF XML, we have the following tags
  - <RDF> </RDF> denote the beginning and end of the RDF description.
  - <Description>'s "about" attribute identifies the subject of the sentence.
  - <Description></Description> enclose the properties and their values.
  - We import Dublin Core conventional properties (creator, title) from outside RDF proper.

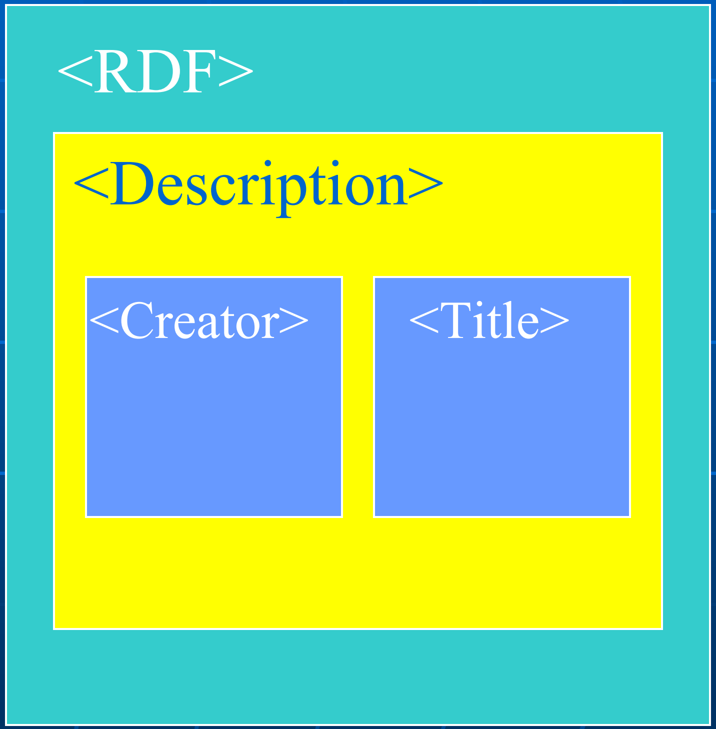
RDF XML: The Gory Details

```
<rdf:RDF
  xmlns:rdf='http://www.w3.org/1999/02/2
  2-rdf-syntax-ns#'
  xmlns:dc='http://purl.org/dc/elements/1.0
  /'>
   <rdf:Description rdf:about='http://.../X‘>
    <dc:creator
      rdf:resource='http://.../people/MEP‘/>
    <dc:title
      rdf:resource='H2O'/>
   </rdf:Description>
</rdf:RDF>
```

Structure of the Document

- RDF XML probably a bit hard to read if you are not familiar with XML namespaces.
- Container structure is illustrated on the right.
- Skeleton is

```
<RDF>
   <Description
   about="">
      <creator/>
      <title/>
   </Description>
</RDF>
```



Aside #1: Creating RDF Documents

- Writing RDF XML (or DAML or OWL) by hand is not easy.
  - It’s a good way to learn to read/write, but after you understand it, automate it.
- Authoring tools are available
  - OntoMat: buggy
  - Protégé: preferred by CGL grad students
  - IsaViz: another nice tool with very good graphics.
- You can also generate RDF programmatically using Hewlett Packard Labs’ Jena toolkit for Java.
  - This is what I did in previous example.

Aside #2: What's a PURL?

- As you may have deduced, RDF's use of URIs implies the need for registering official names.
- A PURL is a Persistent URL/URI.
  - PURLs don't point directly to a resource
  - Instead, they point to a resolution service that redirects the client to the current URL.
  - See www.purl.org
- The PURL for creator currently points to http://dublincore.org/documents/1998/09/dces/#creator, which defines in human terms the DC creator property.

What is the Advantage?

- So far, properties are just conventional URI names.
  - All semantic web properties are conventional assertions about relationships between resources.
  - RDFS and DAML will offer more precise property capabilities.
- But there is a powerful feature we are about to explore…
  - Properties provide a powerful way of linking different RDF resources
    - “Nuggets” of information.
- For example, a publication is a resource that can be described by RDF
  - Author, publication date, URL are all metadata property values.
  - But publications have references that are just other publications
  - DC’s “hasReference” can be used to point from one publication to another.
- Publication also have authors
  - An author is more than a name
  - Also an RDF resource with collections of properties
    - Name, email, telephone number,

vCard: Representing People with RDF Properties

- The Dublin Core tags are best used to represent metadata about "published content"
  - Documents, published data
- vCards are an IETF standard for representing people
  - Typical properties include name, email, organization membership, mailing address, title, etc.
  - See http://www.ietf.org/rfc/rfc2426.txt
- Like the DC, vCards are independent of (and predate) RDF but are map naturally into RDF.
  - Each of these maps naturally to an RDF property
  - See http://www.w3.org/TR/2001/NOTE-vcard-rdf-20010222/

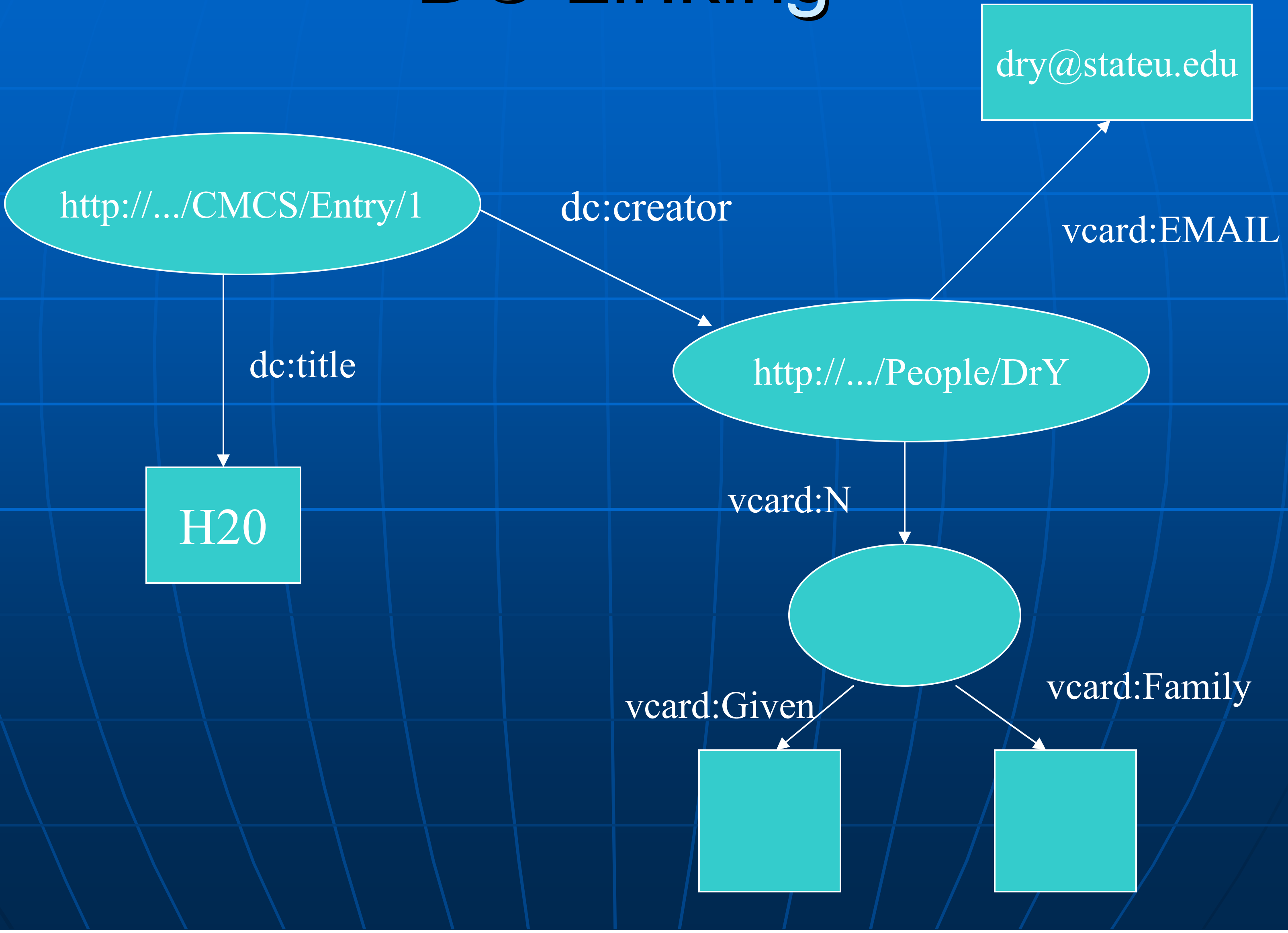
Example: A vCard in RDF/XML

```
<rdf:RDF
   xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
   xmlns:vcard='http://www.w3.org/2001/vcard-rdf/3.0#'>
   <rdf:Description rdf:about='http://cgl.indiana.edu/people/GCF'
       vcard:EMAIL='gcf@indiana.edu'>
      <vcard:FN>Geoffrey Fox</vcard:FN>
      <vcard:N
          vcard:Given='Geoffrey'
          vcard:Family='Fox'/>
   </rdf:Description>
</rdf:RDF>
```

Linking vCard and Dublin Core Resources

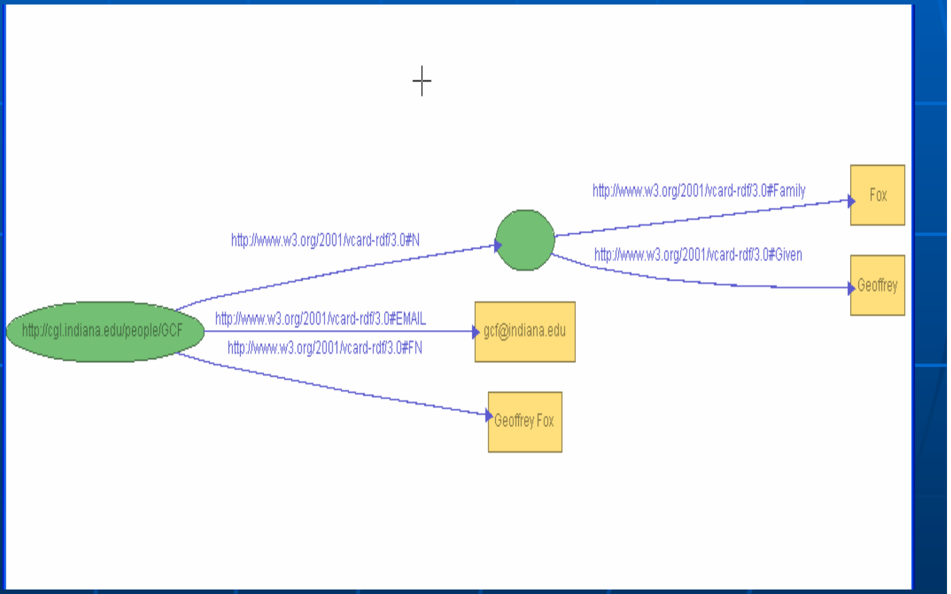
- The real power of RDF is that you can link two independently specified resources through the use of properties.
- We do this using URIs as universal pointers
  - Identify specific resources (nouns) and specifications for properties (verbs)
  - The URIs may optionally be URLs that can be used to fetch the information.
- Linking these resource nuggets allows us to pose queries like
  - “What is the email address of the creator of this entry in the chemical database?”
  - “What other entries reference directly or indirectly on this data entry?”
- Linkages can be made at any time
  - Don’t have to be designed into the system

Graph Model Depicting vCard and DC Linking

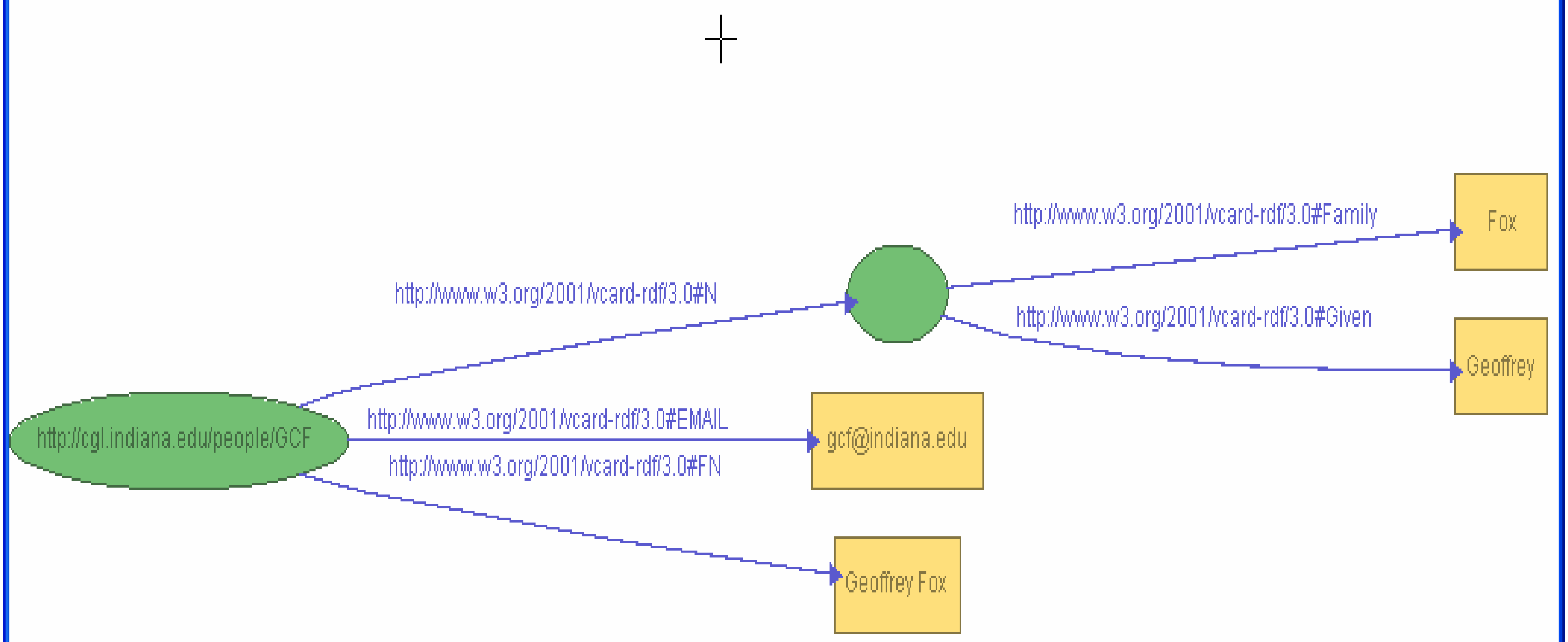


Aside #3: Making Graphs

- IsaViz is a nice tool for authoring graphs.
- Allows you to create and manipulate graphs, export the resulting RDF.
- Graph on right is the vCard RDF from previous slide.

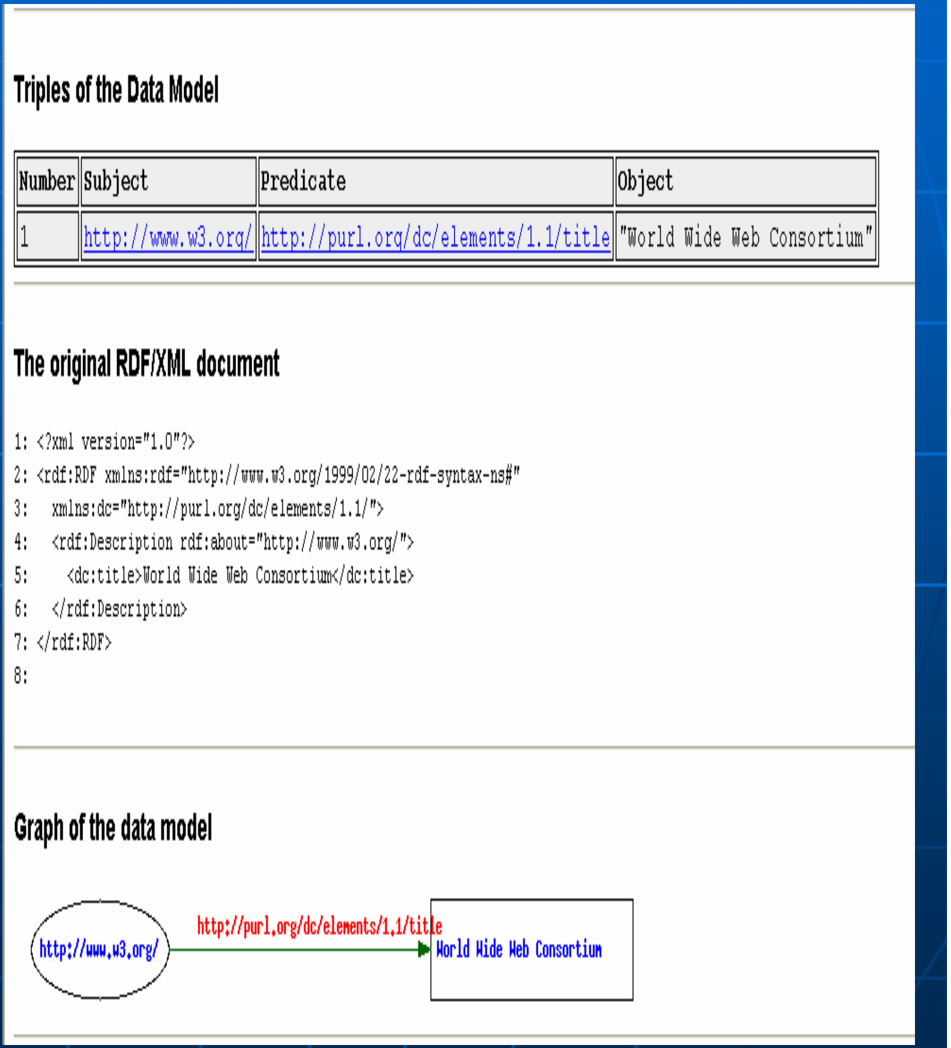

http://www.w3.org/2001/vcard-rdf/3.0#Family
Fox
http://www.w3.org/2001/vcard-rdf/3.0#N
http://www.w3.org/2001/vcard-rdf/3.0#Given
Geoffrey
http://cgl.indiana.edu/people/GCF
http://www.w3.org/2001/vcard-rdf/3.0#EMAIL
gcf@indiana.edu
http://www.w3.org/2001/vcard-rdf/3.0#FN
Geoffrey Fox

IsaViz Screen Shot


http://www.w3.org/2001/vcard-rdf/3.0#Family
Fox
http://www.w3.org/2001/vcard-rdf/3.0#N
http://www.w3.org/2001/vcard-rdf/3.0#Given
Geoffrey
http://cgl.indiana.edu/people/GCF
http://www.w3.org/2001/vcard-rdf/3.0#EMAIL
gcf@indiana.edu
http://www.w3.org/2001/vcard-rdf/3.0#FN
Geoffrey Fox

Another Neat RDF Tool: SiRPAC

- Allows you to parse RDF, convert RDF/XML into graphs and triplets.
- http://www.w3.org/RDF/Validator/



Triples of the Data Model

Number	Subject	Predicate	Object
1	http://www.w3.org/	http://purl.org/dc/elements/1.1/title	"World Wide Web Consortium"

The original RDF/XML document

```
<?xml version="1.0"?>
<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.w3.org/">
    <dc:title>World Wide Web Consortium</dc:title>
  </rdf:Description>
</rdf:RDF>
```

Graph of the data model

http://www.w3.org/
http://purl.org/dc/elements/1.1/title
World Wide Web Consortium

What Else Does RDF Do?

- Collections: typically used as the object of an RDF statement
  - Bag: unordered collection of resources or literals.
  - Sequence: ordered collection or resources or literals.
  - Alternative: collection of resources or literals, from which only one value may be chosen
- And that's about it. RDF does not define properties, it just tells you where to put them.
  - Definitions are done by specific groups for specific fields (Dublin Core Metadata Initiative, for example).
  - RDF Schema provides the rules for defining specific resources classes and properties.

Other Semantic Markup Activities

A tour of other semantic markup language activities.

Other Semantic Markup Languages

- RDF Schema (RDFS):
  - Provides formal definitions of RDF
  - Also provides language tools for writing more specialized languages.
  - We'll examine in more detail.
- DARPA Agent Markup Language (DAML):
  - DAML-OIL is the language component of the DAML project.
  - Defined using RDF/RDFS.
  - We'll examine in more detail.
- Ontology Inference Layer (OIL):
  - OIL language expressed in terms of RDF/RDFS.
  - The OIL project is sponsored by the European Union.
- Web-Ontology Language (OWL):
  - Developed by the W3C's Web-Ontology Working Group
  - Based on DAML-OIL

RDF Schema

- RDF Schema is a rules system for building RDF languages.
  - RDF and RDFS are defined in terms of RDFS
  - DAML+OIL is defined by RDFS.
- Take the Dublin Core RDF encoding as an example:
  - Can we formalize this process, defining a consistent set of rules?
  - Can we place restrictions and use inheritance to define resources?
    - What really is the value of "creator"? Can I derive it from another class, like "person"?
  - Can we provide restrictions and rules for properties?
    - How can I express the fact that "title" should only appear once?
  - Current DC encoding in fact is defined by RDFS.

Some RDFS Classes

RDFS: Resource	The RDFS root element. All other tags derive from Resource
RDFS: Class	The Class class. Literals and Datatypes are example classes.
RDFS: Literal	The class for holding Strings and integers. Literals are dead ends in RDF graphs.
RDFS: Datatype	A type of data, a member of the Literal class.
RDFS: XMLLiteral	A datatype for holding XML data.
RDFS:Property	This is the base class for all properties (that is, verbs).

Some RDFS Properties

subClassOf	Indicates the subject is a subclass of the object in a statement.
subPropertyOf	The subject is a subProperty of the property (masquerading as an object).
Comment, Label	Simple properties that take string literals as values
Range	Restricts the values of a property to be members of an indicated class or one of its subclasses.
isDefinedBy	Points to the human readaable definition of a class, usually a URL.

Sample RDFS: Defining <Property>

```
<rdfs:Class rdf:about="http://.../some/uri">
   <rdfs:isDefinedBy rdf:resource="http://.../some/uri"/>
   <rdfs:label>Property</rdfs:label>
   <rdfs:comment>The class of RDF properties.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="http://.../#Resource”>
</rdfs:Class>
```

- This is the definition of <property>, taken from the RDF schema.
- The “about” attribute labels names this nugget.
- <property> has several properties
  - <label>,<comment> are self explanatory.
  - <subClassOf> means <property> is a subclass of <resource>
  - <isDefinedBy> points to the human-readable documentation.

RDFS Takeaway

- RDFS defines a set of classes and properties that can be used to define new RDF-like languages.
  - RDFS actually bootstraps itself.
- You can express inheritance, restriction
- If you want to learn more, see the specification
  - http://www.w3.org/TR/2003/WD-rdf-schema-20030123/
- But don't trust the write up:
  - Concepts are best understood by looking at the RDF XML. English descriptions get convoluted.
- If you want to see RDFS in action, see the DC:
  - http://dublincore.org/2003/03/24/dces#

What is DAML-OIL?

- RDFS is a pretty sparse
  - Meant to be extended into more useful languages.
- Some missing features, summarized on next table.
- DAML-OIL builds on RDF and RDFS to define several more useful properties and classes.
- DAML-OIL is an assertive markup language for
  - describing resources (labels, comments, etc.)
  - expressing relationships between resources through properties
- It is not a programming language.
  - Compliant DAML parsers and other tools must obey assertion rules.

Some DAML Extensions to RDFS

RDFS	DAML
Treats all literals as strings	Defines other data types (floats, integers, etc.)
Can’t express equivalence between properties or classes,	Several DAML property tags, derived from <equivalentTo>
Can’t express relationships like disjointedness, unions, intersections, complements.	DAML property tags for expressing these relationships
Sequences can’t express cardinality restrictions	Several DAML properties, including <UniqueProperty>
Can’t express inversions and transitiveness	DAML tags for asserting these relationships

What's an Ontology?

- "Ontology" is an often used term in the field of Knowledge Representation, Information Modeling, etc.
- English definitions tend to be vague to non-specialists
  - "A formal, explicit specification of a shared conceptionalization"
- Clearer definition: an ontology is a taxonomy combined with inference rules
  - T. Berners-Lee, J. Hendler, O. Lassila
- But really, if you sit down to describe a subject in terms of its classes and their relationships using RDFS or DAML, you are creating an Ontology.
  - See the HPCMP Ontology example in the report.

Philosophy's Fine, but Can I Program It?

- Yes. The HP Lab's Jena package provide Java classes for creating programs to RDF/RDFS, DAML, and now OWL.
- Several tools built on top of Jena
  - IsaViz, Protégé are two nice authoring tools.
- Also tools for Perl, Python, C, Tcl/TK
  - See the W3C RDF web site.

What is the Difference Between the Semantic Web and Databases?

- Databases can certainly be used to store RDF, DAML, etc, entries.
  - Jena optionally stores RDF models in memory, relational DBs, or XML DBs (Berkeley Sleepy Cat).
- Interesting comparison is between Semantic Web and Database Management Systems
  - Arguably these are two ends of same spectrum
  - DMS represent tight coupling of db software, storage, replication, data models, query services, etc.
    - One organization may control all data
  - SW appropriate for large, loosely coupled systems
    - No centralized control of data
    - Metadata may be directly embedded in the data, may be separate, may be scavenged by roving agents, ….

Semantic Web and Web Services

- Metadata is important for the web.
- Most major software companies (IBM, MS, BEA, Oracle, Sun) are more interested in Web Services.
- Metadata about Web Services is more important than metadata about documents.
  - We need a Semantic Grid, not a Semantic Web
- Semantic Web research focuses heavily on knowledge representation.
  - Logical assertions in DAML-OIL, for example
- There is an opportunity for more research on management of simple but fragmented information nuggets.
  - RDF nuggets scattered across the web, linked with URIs.
  - Sophisticated queries can be made over distributed fragments.

Semantic Web Services as a Trend

"The **real problem** in the Defense Department and in the technology world is **not encoding ontological knowledge** – the real problem is semantic **integration** across the thousands of **databases** and **software** packages."

- Dr. Mark Greaves, DARPA Program Manager, DAML 2004 Program Directions 10/14/03

Developing an Ontology for Earthquake Modeling Codes

An abbreviated example in RDFS.

Motivating Scenario

- We have a collection of codes, visualization tools, computing resources, and data sets that we want to combine in an ontology.
- Instances of the ontology can then be made that describe specific resources.
- After we have built instances, we can pose queries on the data to retrieve values.
  - Values may be structured, so we can do "stepped" queries.
- We thus need to start by grouping together related resources.
- http://grids.ucs.indiana.edu/~maktas/servo/index.html

Group 1: Simulation Codes

- Disloc: calculates surface stress displacements causes by a fault placed in an elastic half-space. Surface data can be either on a grid or on defined scattered points. Can also create InSAR-style surface displacements.
- Simplex: inverts Disloc to estimate fault parameters from observed surface displacements. Surface displacements can be either on a grid or at defined points.
- GeoFEST: does a realistic model of stresses created by a fault. Uses finite element method, realistic material properties.
- AKIRA: Converts a geometry (layers, faults) specification into a finite element mesh. Successive calls refine the mesh. Needed as a helper application for GeoFEST.
- lee2geof: Converts the finite element mesh to GeoFest by associating boundary conditions and material properties with the nodes.
- Virtual California: Based on realistic fault and fault friction models, simulates interacting fault systems.

Group 2:Visualization Codes

- We associate simulation codes with zero or more visualization systems.
  - GMT (General Mapping Tool)
  - IDL
  - RIVA
- In practice, we usually refer to scripts for specific tasks rather than the entire toolkit.

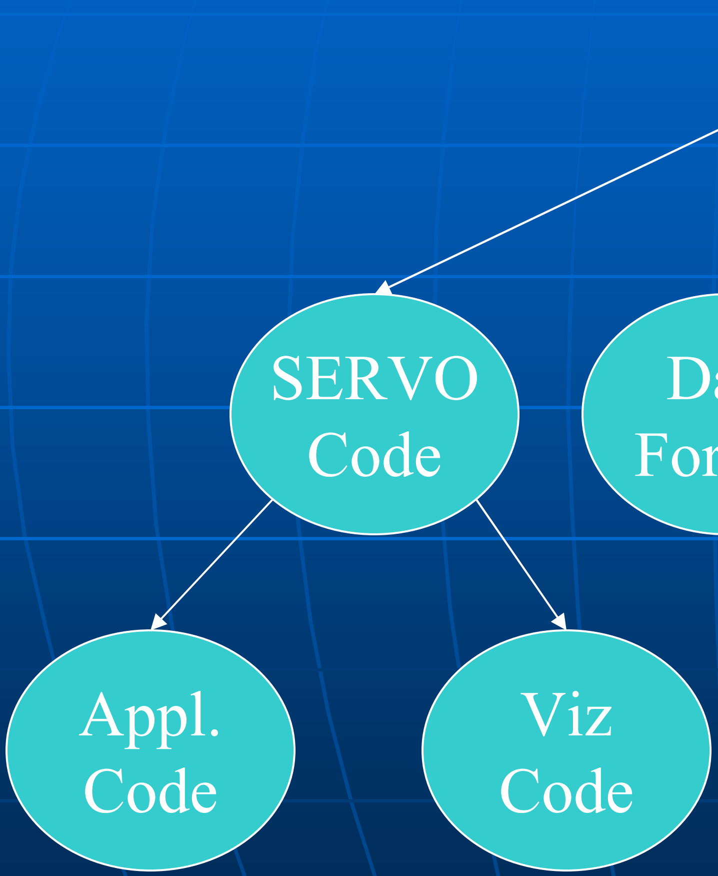
Group 3: Compute Resources

- Grids: a Sun Ultra 60 with Disloc, Simplex, and VC installed.
- Danube: a linux dual processor machine with GeoFEST, lee2geof, Akira, GMT installed.
- Jabba: an SGI 8 processor machine with RIVA installed.

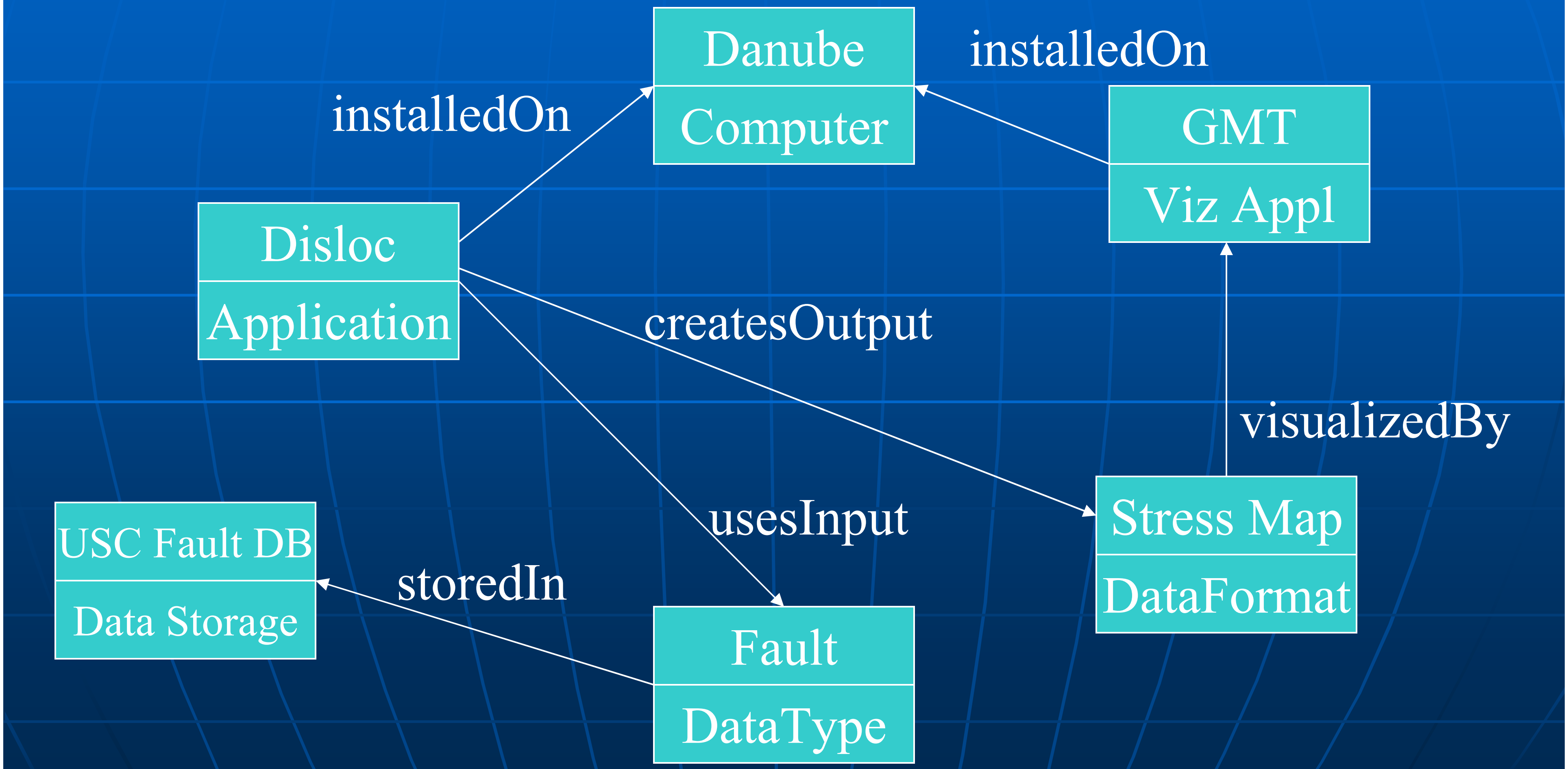
Group 4: Data Types and Formats

- This is a mixture of data objects and representations. As always, the data itself is not represented but information like the creator of the data is.
  - Faults
  - GPS data
  - Seismicity
  - Surface stress data
  - INSAR data
  - Surface data representation: grid or point data

Some SERVO Metadata Objects Inheritances



Some Sample Relationships



Front Matters

- We now wish to create classes and properties that we can couple into an ontology.
- First, let's define a base object, GEMObject, that we will extend as necessary.
- This object doesn't do anything but it will have some uses when we define property ranges and domains.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdfs=http://www.w3c.org/2000/01/rdf-schema#>
   <rdf:Description rdf:ID="GEMObject">
    <rdf:type="http://www.w3c.org/2000/01/rdf-schema#Class"/>
    <rdfs:label>GEMObject</rdfs:label>
    <rdfs:comment>This is a generic object from which everything in our
    ontology will be derived.
    </rdfs:comment>
   </rdf:Description>
</rdf:RDF>
```

<Description>

- The <RDF> tag is followed by the <Description> tag. This serves two important purposes:
- A <Description> surrounds the property and values for a class.
- It also identifies the "thing" that the description applies to.
- The "thing" is actually a resource and is identified by either a local or absolute URI.

Defining Some Useful Classes

- Based on our introductory comments, we need the following classes:
  - GEMCodes, with “application” and “visualization” extensions
  - GEMData, such as Faults, GPS, and so on.
  - GEMDataFormat: either grid or point data
  - ComputeResources: host computers.

Example: Defining a GEMCode

- GEMCodes should extend our GEMObject generic superclass.
- It should itself be extended by other, more specific resource types.

```
<rdf:Description rdf:ID="GEMCode">
   <rdf:type="http://www.w3c.org/2000/01/rdf-
   schema#Class"/>
   <rdfs:subClassOf rdf:resource="#GEMObject"/>
   <rdfs:label>GEMCode</rdfs:label>
   <rdfs:comment>This is a general code class that we will
   extend</rdfs:comment>
 </rdf:Description>
```

Defining Properties

- Classes by themselves don’t tell us much, but when we associate them with properties, things start to fall into place. Before describing how a property may be encoded, let’s try and enumerate the ones that we will need.
  - ownsGEMResource: who owns a particular resource.
  - installedOn, hasCode: where a GEMCode is installed, or, conversely, what codes a particular computing resource has.
  - hasData: where some piece of data is (on some compute resource).
  - hasDataFormat: associate a data format with a piece of data.
  - takesInputData, createsOutputData: what kind of data a code takes as input and generates as output.
  - dependsUpon: a code depends upon another operation before it can be completed.

A Property for Resource Ownership

Now let’s look at how to encode this first property. It looks like this:

```
<rdf:Description rdf:ID="ownsGEMResource">
    <rdf:type
    rdf:resource="http://www.w3c.org/1999/02/22-
   rdf-syntax-ns#Property"/>
   <rdfs:domain
   rdf:resource="http://www.w3c.org/2001/vcard-
   rdf/3.0#"/>'
   <rdfs:range rdf:resource="#GEMObject"/>
</rdf:Description>
```

More Information

- The W3C Semantic Web Activity:
  - http://www.w3.org/2001/sw/
- The Dublin Core Metadata Initiative:
  - http://dublincore.org/
- For some survey reports:
  - http://www.servogrid.org/slide/GEM/SW
  - See these reports for longer discussions, examples, and references.
- For programming examples, see http://www.servogrid.org/slide/GEM/SW/Examples
  - Written in Java, using HPL’s Jena 1.6.1
  - Jena 2 is available, haven’t checked backward compatibility.

Tools for Playing with Things

- Jena Toolkit: Java packages from HPLabs for building Semantic Web applications.
  - http://www.hpl.hp.com/semweb/
  - Both IsaViz and Protégé use this.
- IsaViz: A nice authoring/graphing tool
  - http://www.w3.org/2001/11/IsaViz/
- Protégé: Another ontology authoring tool
  - http://protege.stanford.edu/

Survey Reports at www.servogrid.org/slide/GEM/SW

Semantic Web.doc	Surveys RDF, RDFS, DAML, and OWL
Semantic Web II.doc	Surveys major Semantic Web software development efforts and products.
Semantic Web IIIA.doc	Tutorial material for authoring RDF and RDFS.
Semantic Web IIIB.doc	Tutorial material for authoring DAML-OIL.
Tool Review—Jena.doc	Programming tutorial using Jena.
Tool Review—Ontomat	User's survey for OntoMat.
Tool Review--Protege	User's survey for Protégé.

Programming Examples

DublinCoreEx1.java	Demonstrates how to create a DC model for describing publications.
DublinCoreEx2.java	Demonstrates reading and writing RDF files.
DublinCoreEx3.java	Demonstrates union, intersection, and difference operations on two models.
DublinCoreEx4.java	Converts a bibtex file into a DC model.
DublinCoreEx5.java	Extends Ex4 to create vCards to link DC and VCARD resources.
DublinCoreEx6.java	Simple DC+VCARD example.
DublinCoreEx7.java	Extends Ex5 to uses RDQL queries