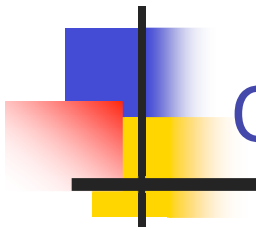
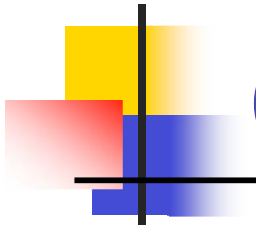


The Semantic Web

Ontology Development Using RDF and OWL

Hari Rajagopal

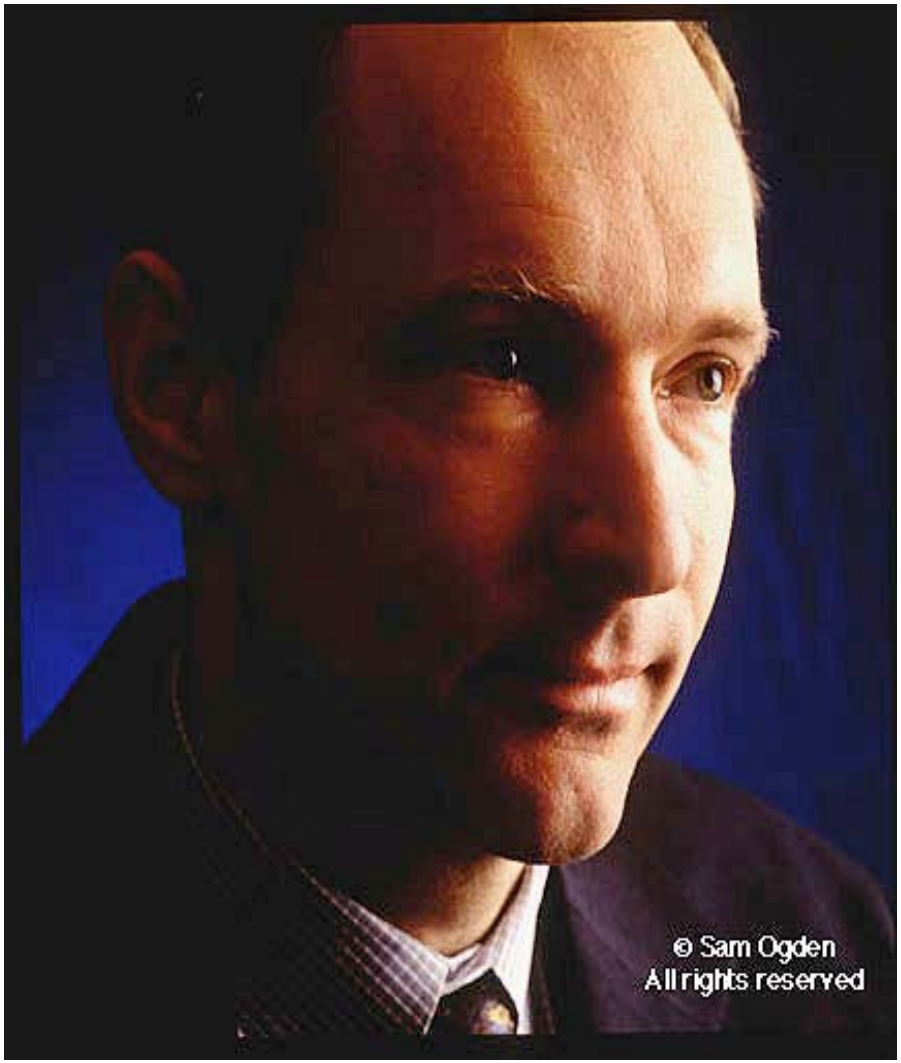
IBM Corporation

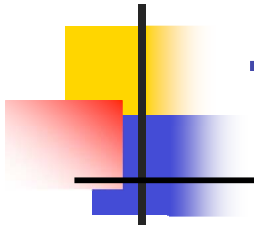


Contact Info

- Hari Rajagopal
  - harir@us.ibm.com

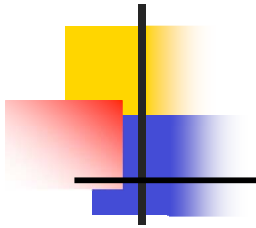
The World Wide Web – Quiz





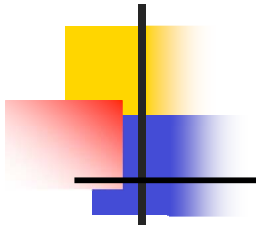
The Web Today

- Estimated billion pieces of static information
- Constantly being
  - Searched by applications and people
  - Updated by people
  - Referred to by documents
  - Extracted from by humans



Problems?

- Keyword based searches bring up pure matches and miss out on context
- Human intervention is needed in order to summarize and integrate information from diverse sources
- Updating and refreshing of data is very complex, needs human intervention
- Machine readability is a major issue

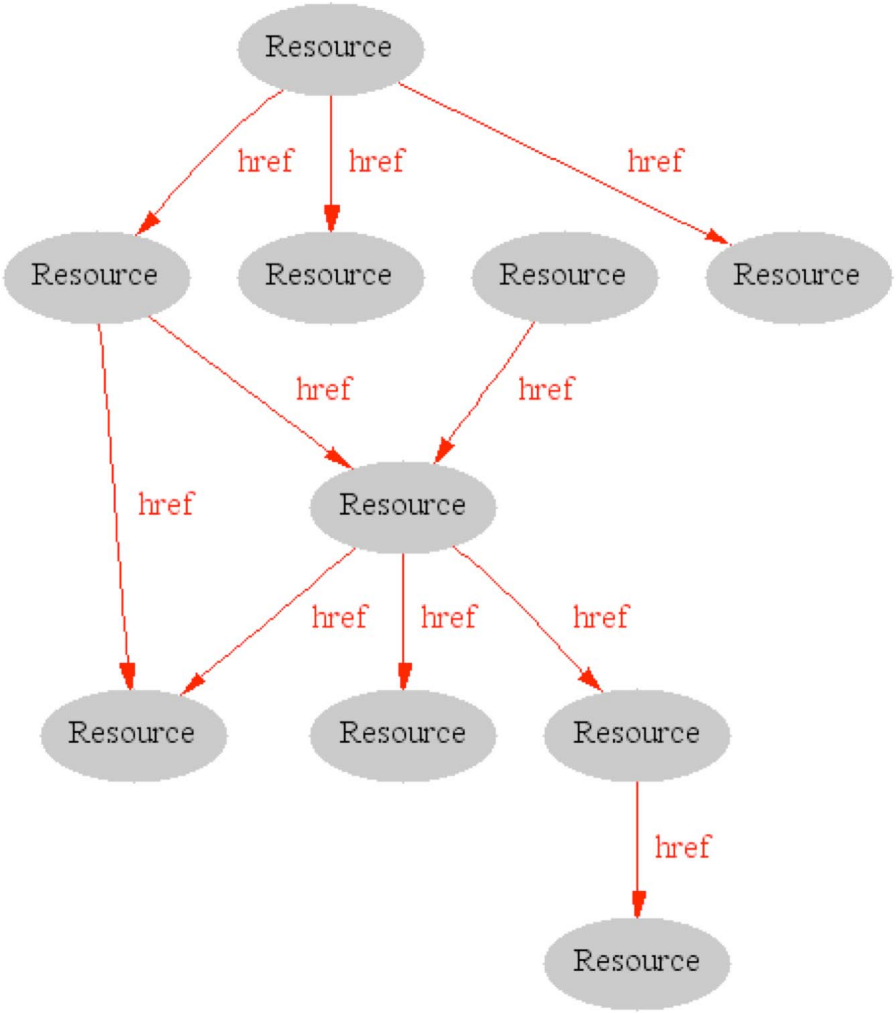


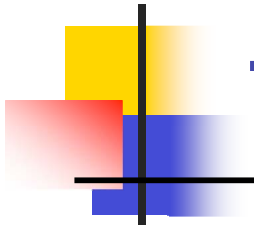
Example

- Let us digress and do a search for a piece of information on the web
- Some are very specific, some are not

Some Definitions

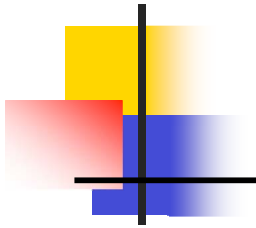
- The web today is syntactic in nature
- Navigation, search and making sense of items is *via* links from resource to resource (hard links)



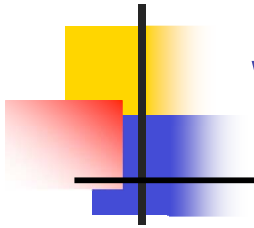


Today – Computers Can't Do

- Linking, rationalization and adding meaning to data
  - That has to be done by people, and then displayed and presented by computers (still glorified decision makers and rendering engines)

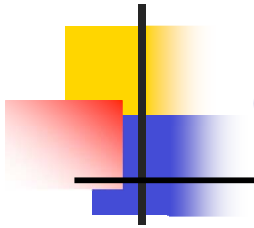


- ***Metadata is the magic bullet !!!***



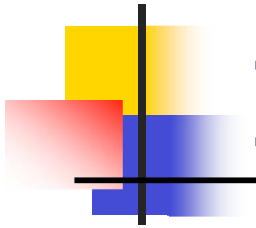
What Is Metadata?

- Very clear in some contexts – Databases for instance (no pun intended)
- Not so clear in others
- The take home message – metadata is whatever is relevant to the domain



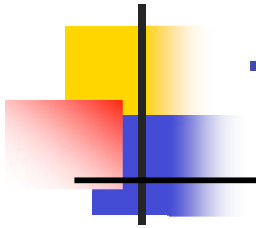
Or Not!

- Some attempts at using metadata
  - The <meta> tag in the HTML header
    - Applies to entire doc
    - Can't scope to specific resources within it
      - Can do so using anchor tags, but non-standard



Information Representation

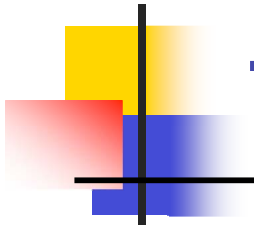
- Common language has been arrived at
  - XML and its derivatives
  - Structured, easily parsable
- Does this make it better understood?
- Or simply more interoperable?



The Semantic Web

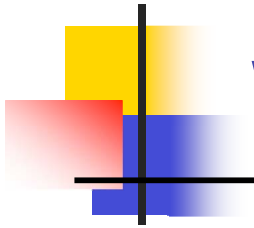
- So, what is the Semantic web ?

- "*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.*" – *Tim Berners-Lee*



The Semantic Web *(Continued)*

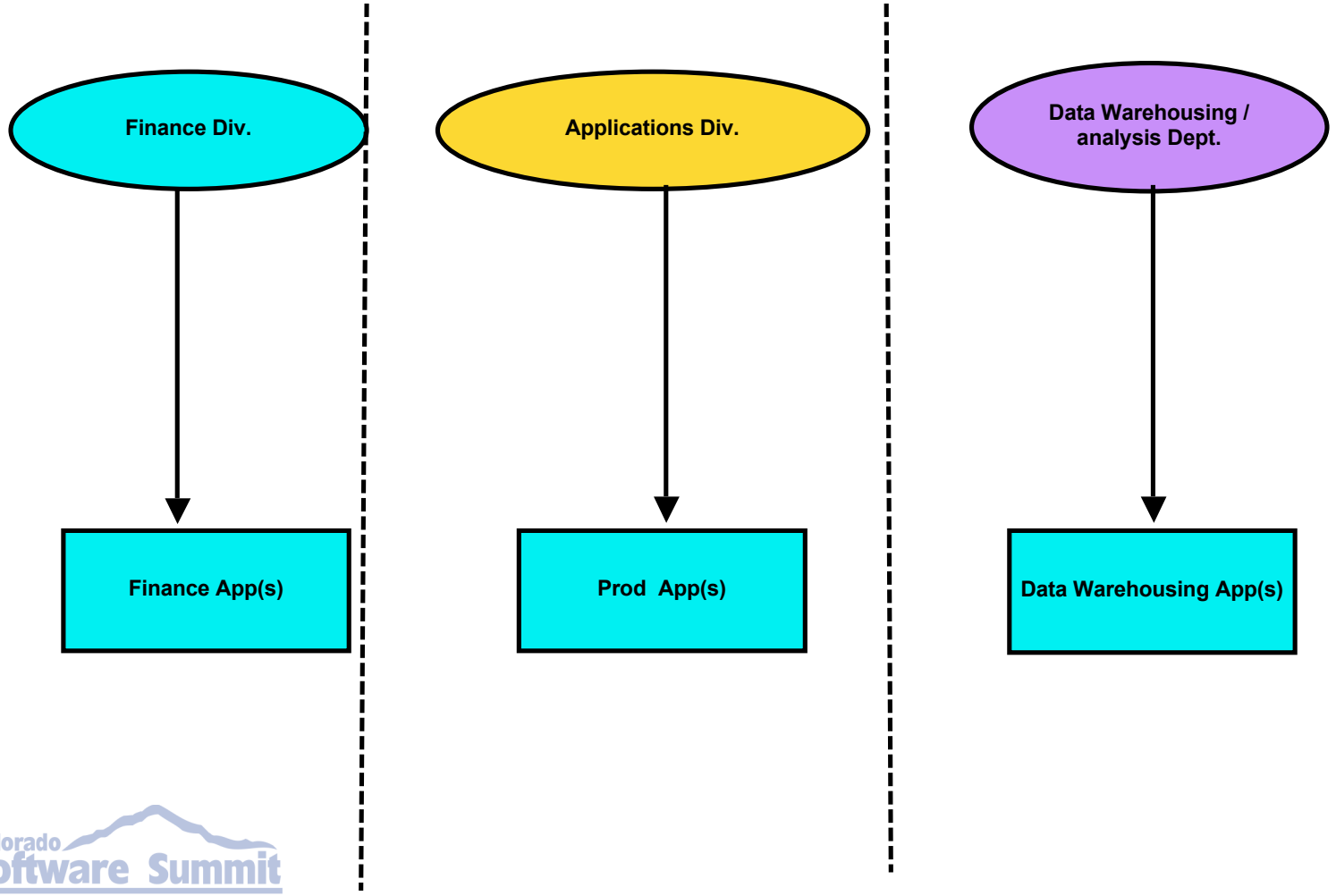
- What the Semantic web is NOT
  - Artificial Intelligence
  - Inference system

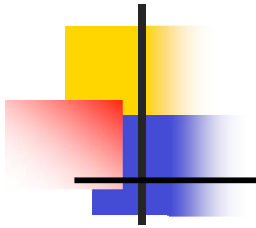


Within an Enterprise…

- XML needs schemas to mean something
  - Otherwise – basically config files, or EDI
- Schemas fill the gap here, sometimes
  - Across departments within an enterprise they don't always align
  - Hence the need for a enterprise domain model

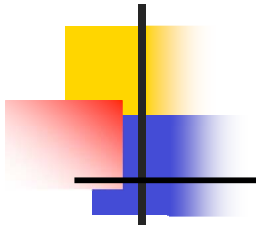
Schemas in Silos





Problems

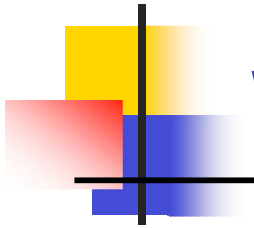
- Issues with sharing schemas
  - Could be resolved using a common enterprise wide domain model
  - Might not work in a cross-enterprise fashion
- Issues with searching/usage due to lack of commonality



Problem at Hand

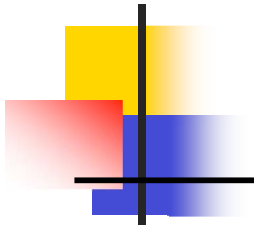
- People want to find meta-data about the subject of interest

- Whether it is a critique of a book, a tidbit of info about a specific dog breed or a flavor characteristic of a particular wine, it is the MetaData (data about data) that is of interest



What Would This Be Used For?

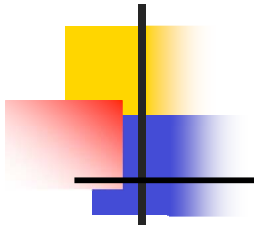
- Satisfy curiosity (book annotations)
- Find a pet (dog breed behaviour specifics)
- Monitor who's checking out a book (FBI)
- List goes on…



Searching for Data

- The web has an incredible wealth of resources (note the term used here) that are accessible *via* URL(s)

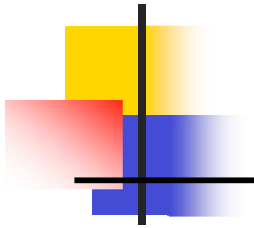
- Most of these searches though are based on crawlers, robots, spiders and the like
- Not contextual data but text based data
- There are a couple of exceptions to this norm – Yahoo!



How Does Yahoo Work ?

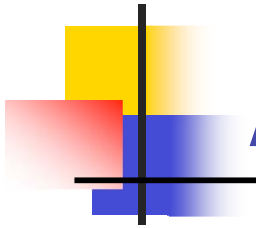
- The answer is 'people power' ☺
- People classify, categorize and rank resources and make them web accessible
- Akin to the monks of old sitting in a cathedral monastery copying and classifying books (see *The Name of the Rose* by Umberto Eco)

The Yahoo Cataloguer at Work



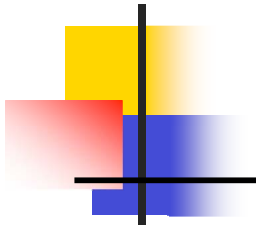
People Still Need…

- A way to represent the data
- A language to describe data items
- A means of linking the terms (vocabulary) of the language into meaningful relations



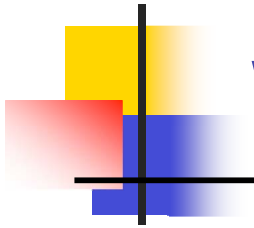
Any More Requirements?

- A system of machine-processable identifiers for identifying a subject, predicate, or object in a statement without any possibility of confusion with a similar-looking identifier that might be used by someone else on the Web
- A machine-processable language for representing these statements and exchanging them between machines.



RDF

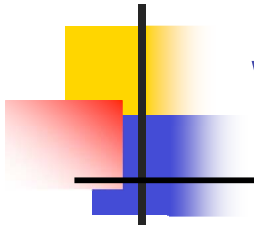
- Enter the Resource Description Framework



What Is RDF?

- The Resource Description Framework (RDF) is the W3C standard that is the foundation for the Semantic Web

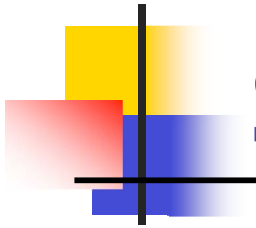
- It provides a way to describe metadata about information in an interoperable, machine-readable format



What Is RDF? *(Continued)*

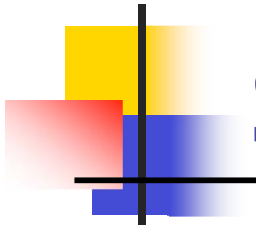
- RDF is simply a way of asserting facts using a vocabulary that is specified in XML schema

- It facilitates assertion and processing of these facts in an automatable fashion (Machine processing *vs* Human readable)



Some Basics

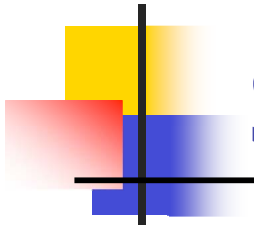
- Resources
  - An entity that has a valid URI
- Properties
  - A Resource that has a name and can be used as a property
- Statements
  - An assertion linking the above together



Some Basics *(Continued)*

- Statements apply a value to a specific property of a resource

- For example:
  - In a web-ring for mastiff owners, the entry –
    - The Molosserworld website has a site address of http://www.moloss.com
    - The resource (Molosserworld website) has a property (URL) and a value for that (the HTTP address)

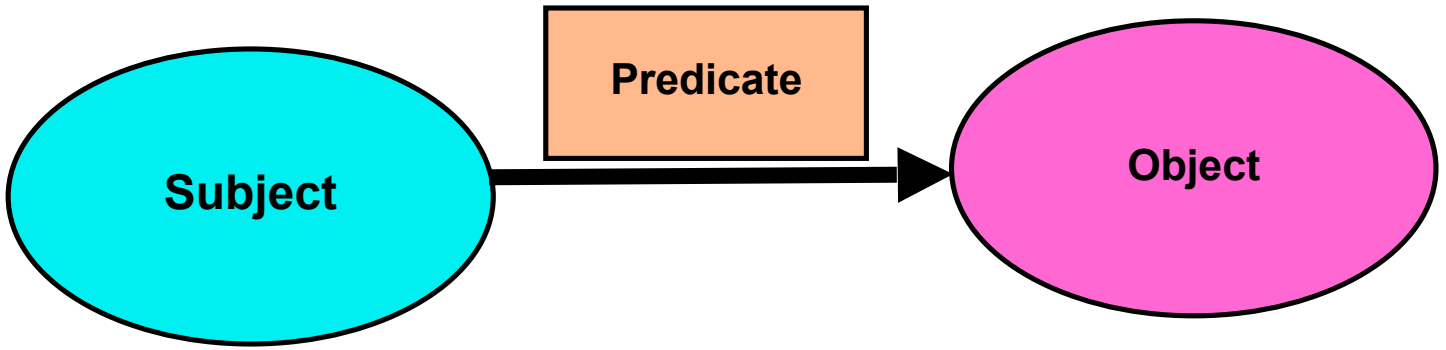


Some Basics *(Continued)*

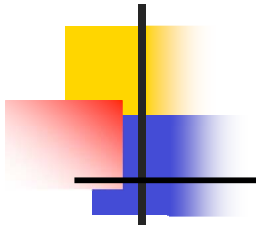
- These map to the Subject, Object and predicate of formal logic.

- Here the Subject is the 'link', the predicate is 'has URL address' and the Object is a resource (could be a literal) with the website address (http://www.moloss.com)
- Note: the Object could be a string literal too

Assertions – Typical Format



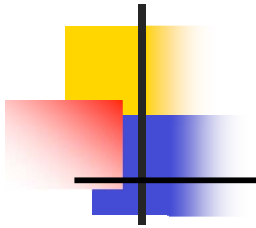
- If an Object is to have properties of its own, then it needs to be a URI, otherwise it can be simply a string literal
- Bottom line – *Anything can be a resource*



RDF Has Rigor

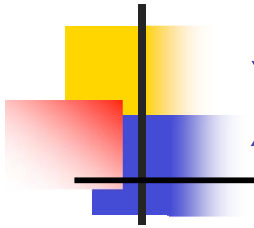
- The above explanation of the assertion seems very loosely put

- RDF constrains it and imposes restrictions on it
  - The subject MUST be a resource
  - A resource in RDF terms boils down to a URI
  - Objects can be literals or resources themselves



Representation

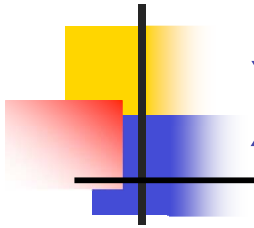
- RDF is based on the principle of a 3-tuple
- Representation of a tuple:
  - (Resource, Property, Value)
  - Tuples from one model can be mapped to another, makes *Federation* of data possible
- Format for RDF is well known and understood, namely XML
  - So why not use XML itself, why another format?



XML – Pros and Cons

- XML is the obvious choice for Data Format for information exchange

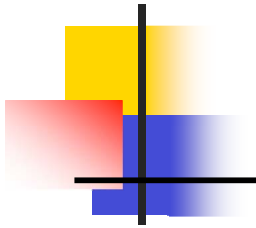
- XML falls short in some respects:
  - Processing power needed for parsing
  - Scalability (processing tuples is much easier)
  - Order of elements is important
- Thus not the best choice for Metadata, but a great alphabet for writing it



XML – Pros and Cons *(Continued)*

- RDF data can be expressed using an XML syntax, allowing it to be passed over the Internet as a document and parsed using existing XML based software

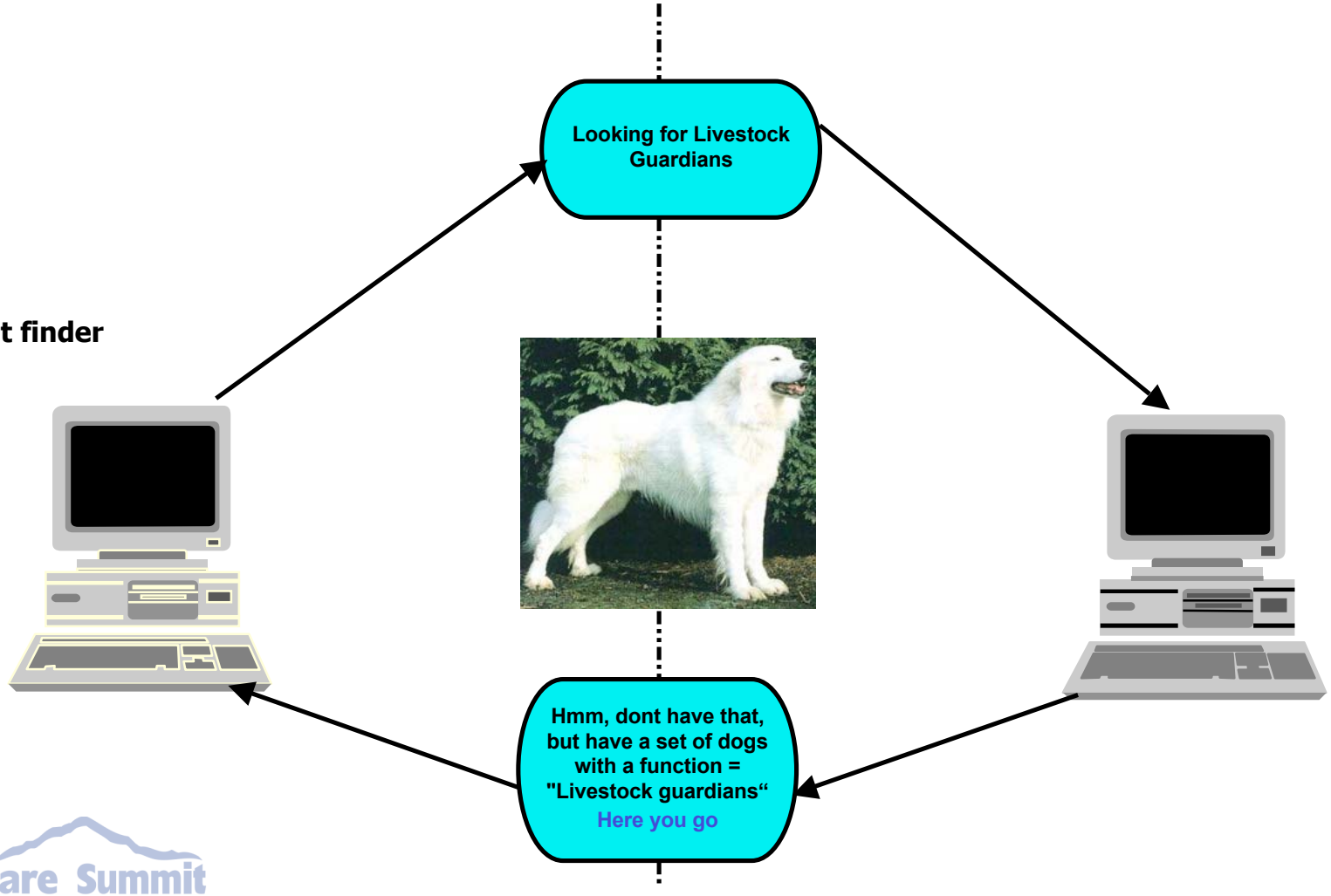
- Once again – XML is a great alphabet but has no meaning and does not represent relationships very well

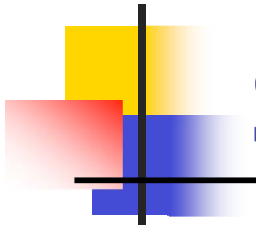


What does all this mean?

Scenario

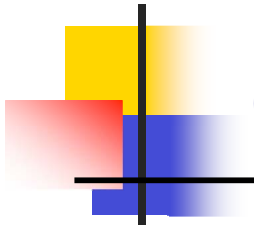
Online pet finder





Scenario *(Continued)*

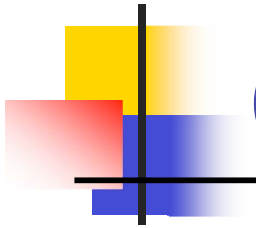
- In the preceding example
  - One source of data was queried by another
  - The syntax used by the client was understood by the receiver
  - The syntax was mapped to the receiver's knowledge base
  - A response was generated by combining knowledge representations



Case Study

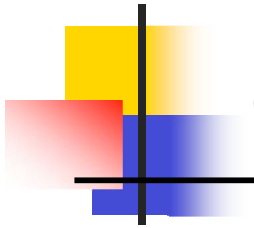
- The AKC (Armenian Kennel Club) and the FCI (Federation Cynologie Internationale) want to collaborate and design a knowledge base for dog breeds

- Must be searchable using loosely defined criteria, easily describable and accessible *via* the web (of course ☺)



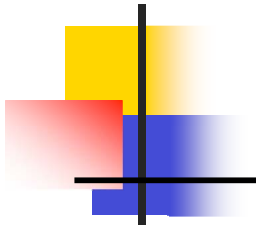
Case Study *(Continued)*

- Suppose we want to represent the following:
- A Pastor Abruzzi is a working dog from Italy that functions as a livestock guardian
- We see the two resources immediately:
  - Italy (there may be more info associated with it)
  - Pastor Abruzzi (the specific breed of dog, Object)



Case Study *(Continued)*

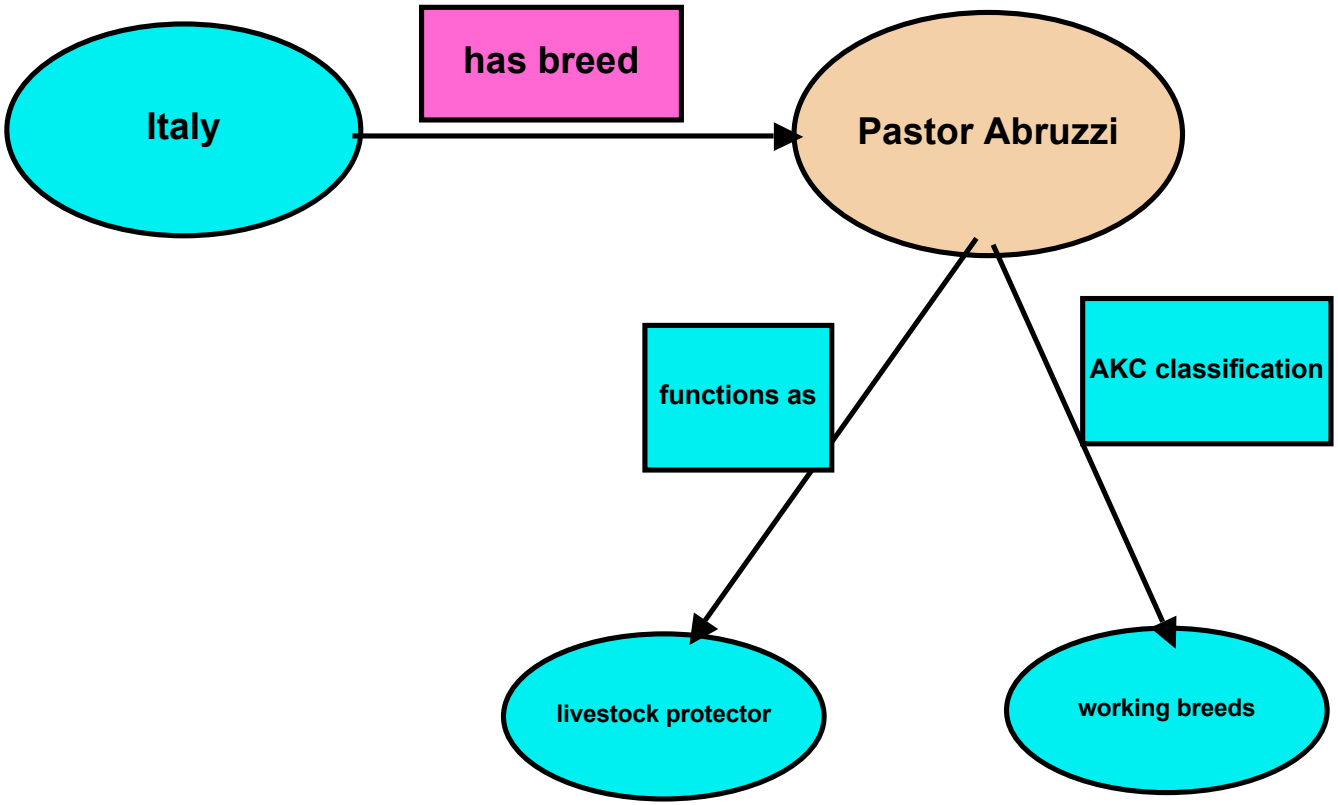
- A couple of predicates show up as well:
  - Functions
  - Native-to
  - Working dog (maybe – as a boolean property)



Perspectives

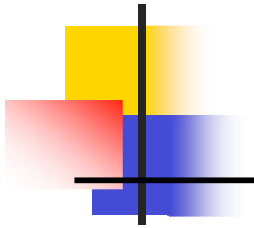
- One could look at the problem from the angle of classifying dogs according to their function
- Another approach would be breed size
- A third might be the country of origin
  - Any more??

World of Dogs



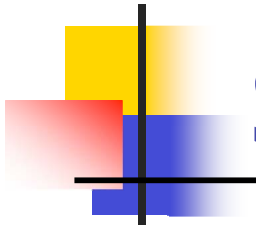
XML Representation

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns="http://schemas.cssexample.org/rdfexample/">
  <rdf:Description about="http://schemas.cssexample.org/Italian-Breeds">
    <includes>
      <rdf:Description ID=""Pastor_Abruzzi">
        <AKC_classification>Working_Breed</AKC_classification>
        <function>Livestock_guardian</function>
      </rdf:Description>
    </includes>
  </rdf:Description>
</rdf:RDF>
```



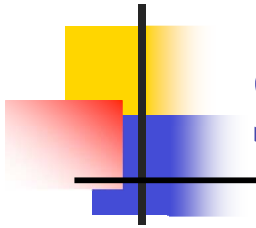
Salient Points

- Note the use of rdf:RDF to wrap the statements
- Namespaces are key
- The subject is referred to by the URI ../Italian-Breeds
- The only predicate for this here is the <includes>



Salient Points *(Continued)*

- The object here (pastor-abruzzi with a URI) acts as a subject and has predicates of its own:
  - Function
  - AKC-Classification
- Both of the values(objects) for these predicates are string literals

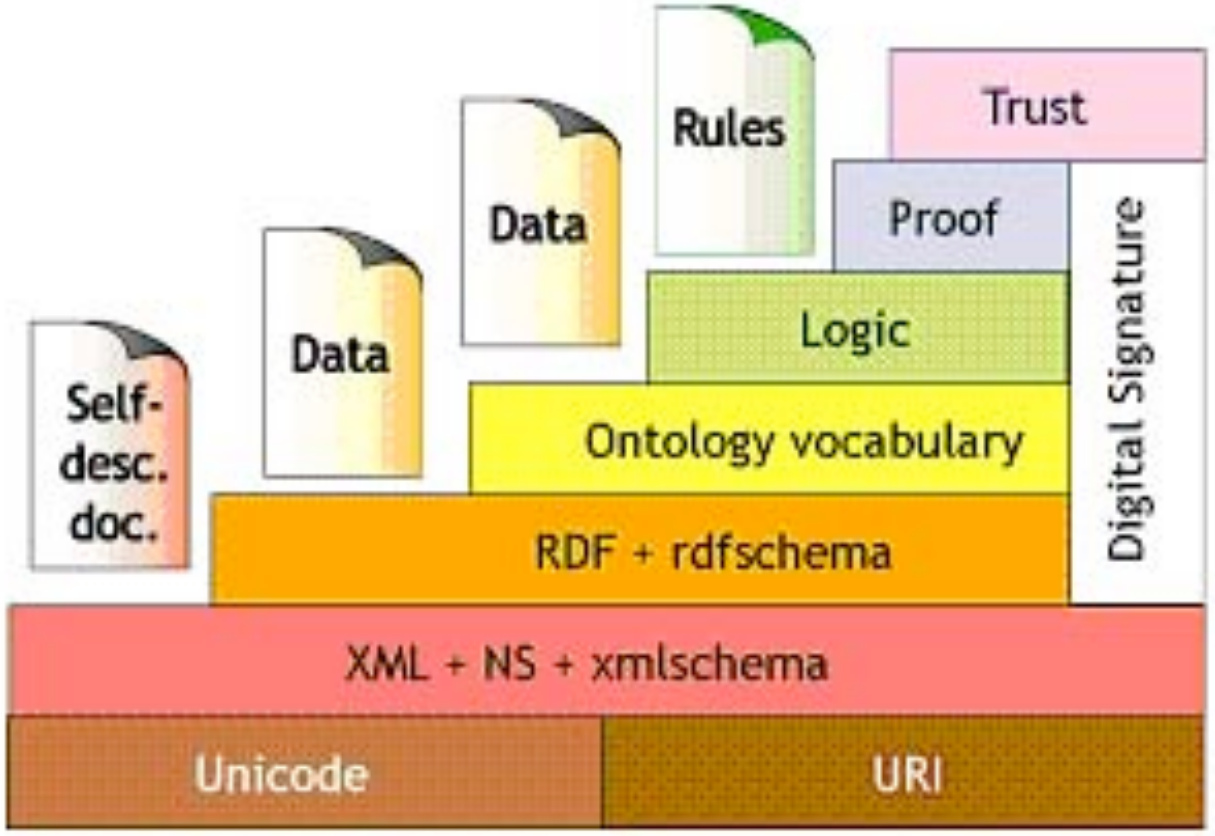


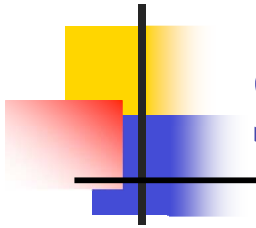
Salient Points *(Continued)*

Ladies and Gentlemen,

What we have here is the basis of a model

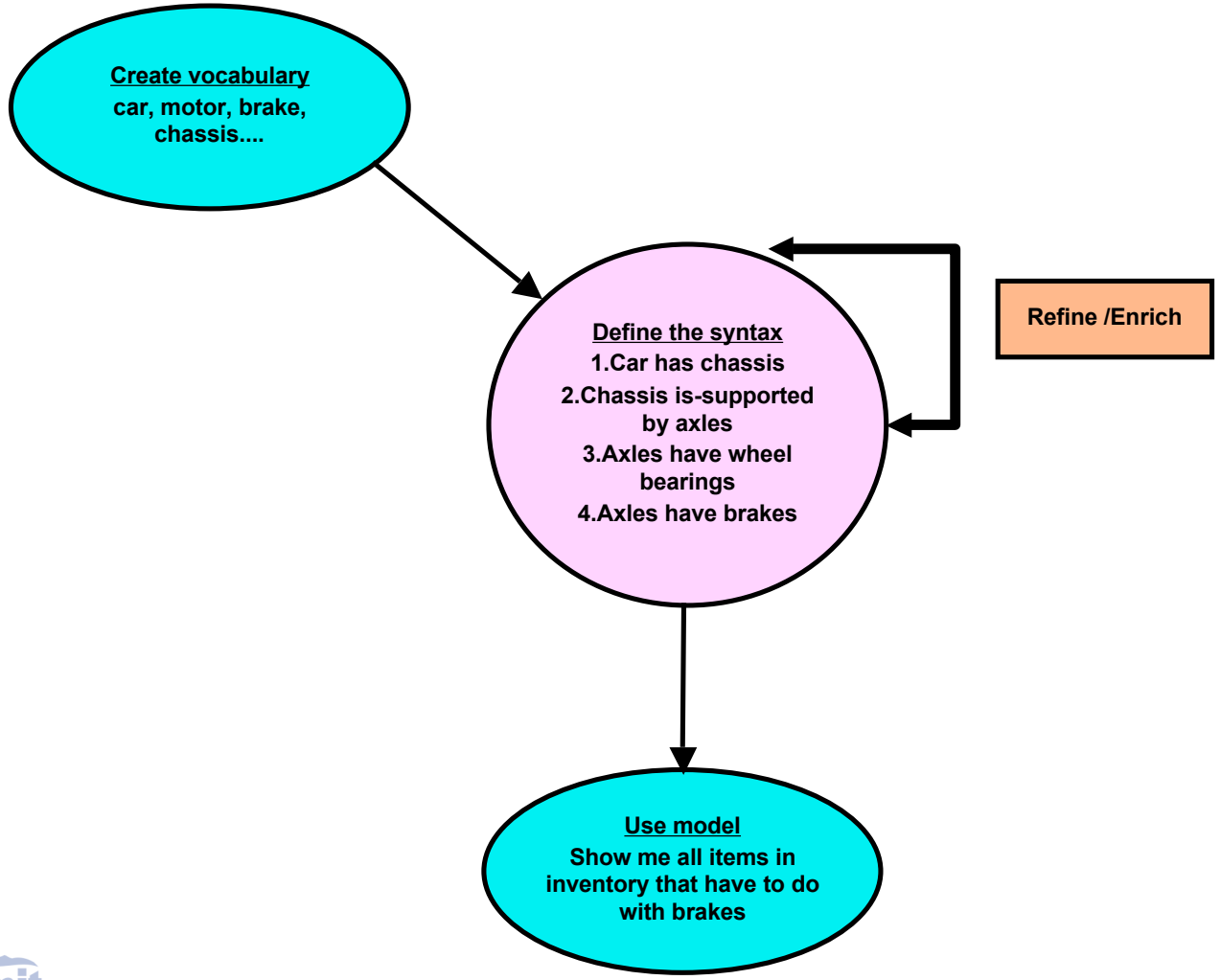
Semantic Web Stack

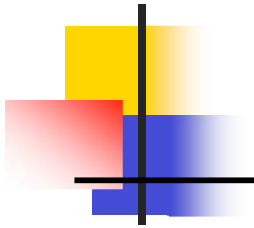




Steps in Defining a Model

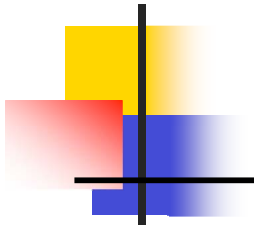
- Create the vocabulary using XML Schema
- Create the Syntax using RDF
- Create RDF triples that express the relationships
- Refine the above
- Enrich the model





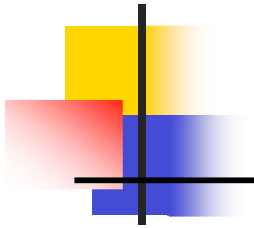
Steps in Creating a Model

- Identify a resource within the system
- Create a vocabulary
- Define properties for the system/domain
- Link resources to properties and their values
- Refine iteratively



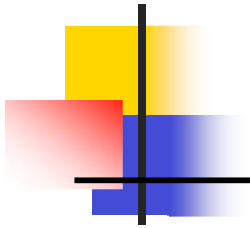
Defining the Model

- Ontological definition – rules on top of the model
- Ontology – "*An ontology is a specification of a conceptualization*"
- A simpler definition – "*representation of knowledge about a particular domain'*



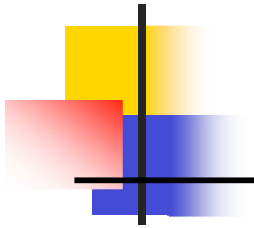
Schemas *vs* Ontologies

- XML schemas specify the alphabet for a vocabulary (RDF statements)
- Ontologies use that vocabulary and lay out grammatical rules and relationships
  - Analogous to the alphabet and a language
  - Can be represented as a collection of RDF-triples



Where Is RDF Used Today ?

- RSS feeds
- Search engines
- Ontological searches and information cataloguing
- Online search engines are moving to metadata based searches



RSS

- RDF Syndicated Source
- Based on RDF
- Uses RDF syntax

- If it was plain XML w/o the RDF it would be much less topic specific and much harder to zone in on subjects (note the term ☺) of interest

Sample RDF from Craigslist.org

```
<rdf:RDF>
 - <channel rdf:about="http://www.craigslist.org/car/">
    - <title>
        craigslist | cars & trucks in san francisco bay area
      </title>
      <link>http://www.craigslist.org/car/</link>
      <description/>
      <dc:language>en-us</dc:language>
      <dc:rights>Copyright 2005, craigslist.org</dc:rights>
      <dc:publisher>webmaster@craigslist.org</dc:publisher>
      <dc:creator>webmaster@craigslist.org</dc:creator>
      <dc:source>http://www.craigslist.org/car/</dc:source>
    - <dc:title>
        craigslist | cars & trucks in san francisco bay area
      </dc:title>
      <dc:type>Collection</dc:type>
      <syn:updatePeriod>hourly</syn:updatePeriod>
      <syn:updateFrequency>4</syn:updateFrequency>
      <syn:updateBase>1901-01-01T00:00+00:00</syn:updateBase>
      <dcterms:tableOfContents>http://www.craigslist.org/car/</dcterms:tableOfContents
      <dcterms:isFormatOf>http://www.craigslist.org/car/</dcterms:isFormatOf>
    - <items>
       - <rdf:Seq>
            <rdf:li rdf:resource="http://www.craigslist.org/sby/car/94434598.html"/>
            <rdf:li rdf:resource="http://www.craigslist.org/nby/car/94434533.html"/>
            <rdf:li rdf:resource="http://www.craigslist.org/nby/car/94434486.html"/>
```

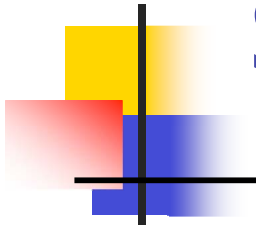
Sample RDF from Craigslist.org *(Continued)*

- So, what's the 'dc:' namespace prefix shown in the previous example ?

- It refers to the Dublin Core group
  - An attempt at a common set of RDF terms
  - So, if we are talking about a 'dog' using dc:dog unmistakably tells us that it is a canine we are referring to and not Snoop Doggy …

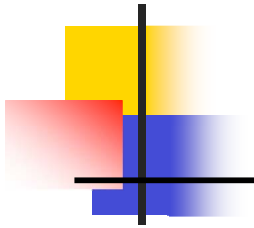
Sample RDF from Craigslist.org *(Continued)*

```
<item rdf:about="http://www.craigslist.org/nby/car/94434486.html">
  <title>1955 Ford stepside (Santa Rosa) $3500</title>
  <link>http://www.craigslist.org/nby/car/94434486.html</link>
  <dc:rights>Copyright 2005, craigslist.org</dc:rights>
  <dc:source>http://www.craigslist.org/nby/car/94434486.html</dc:source>
  <dc:language>en-us</dc:language>
  <dc:date>2005-08-30T16:22:48-07:00</dc:date>
  <dc:title>1955 Ford stepside (Santa Rosa) $3500</dc:title>
  <dc:type>text</dc:type>
  <dcterms:issued>2005-08-30T16:22:48-07:00</dcterms:issued>
</item>
```



Sample RDF from Craigslist.org
(Continued)

- This is how instead of the traditional robots and crawlers we can now have efficient syndication of content using the RDF based schemas available
- We can write an app to scrape craigslist for all Ford trucks that have a certain mileage and are located in a specific zone of California



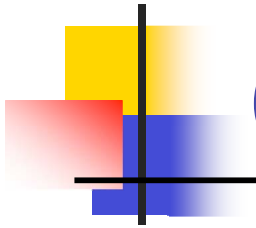
Deeper into RDF

- With RDF one can go a step further and define classes and subclasses to further enrich the model

```
<rdfs:Class rdf:ID="carPart">
</rdfs:Class>
```

Case Study www.myfurniture.com

- Capture knowledge base
  - Designer input
- Create Model vocabulary
- Create relationships
- Query model
- Refine model

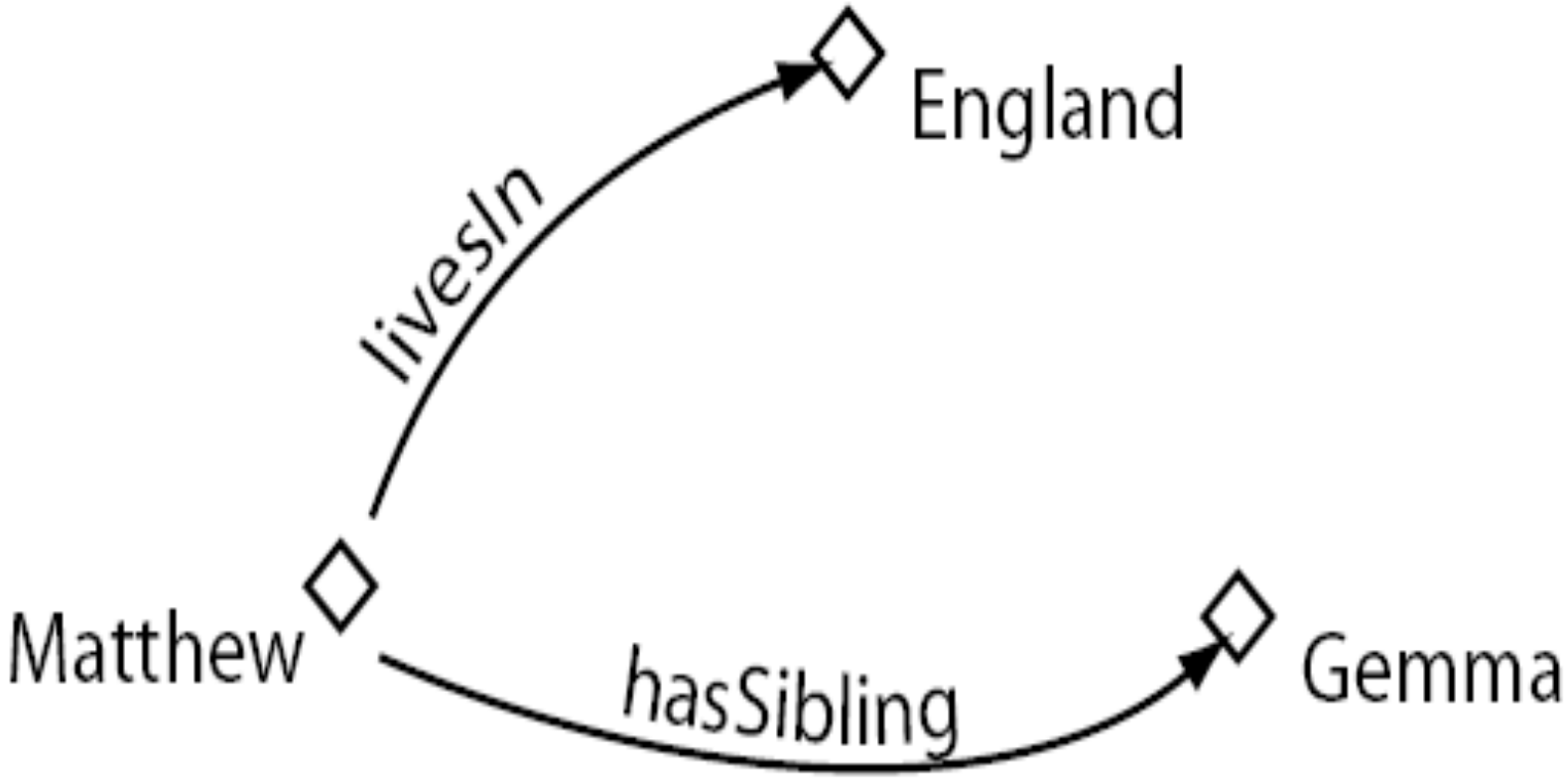


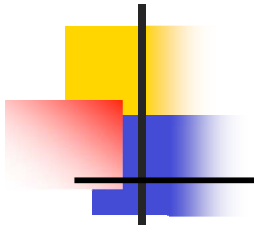
Ontology Model

- An ontology model is an extension of the RDF model that provides extra capabilities for handling ontology data sources

- Some terms are different
  - Properties (*aka* predicates)
  - Individuals
  - Classes

Properties

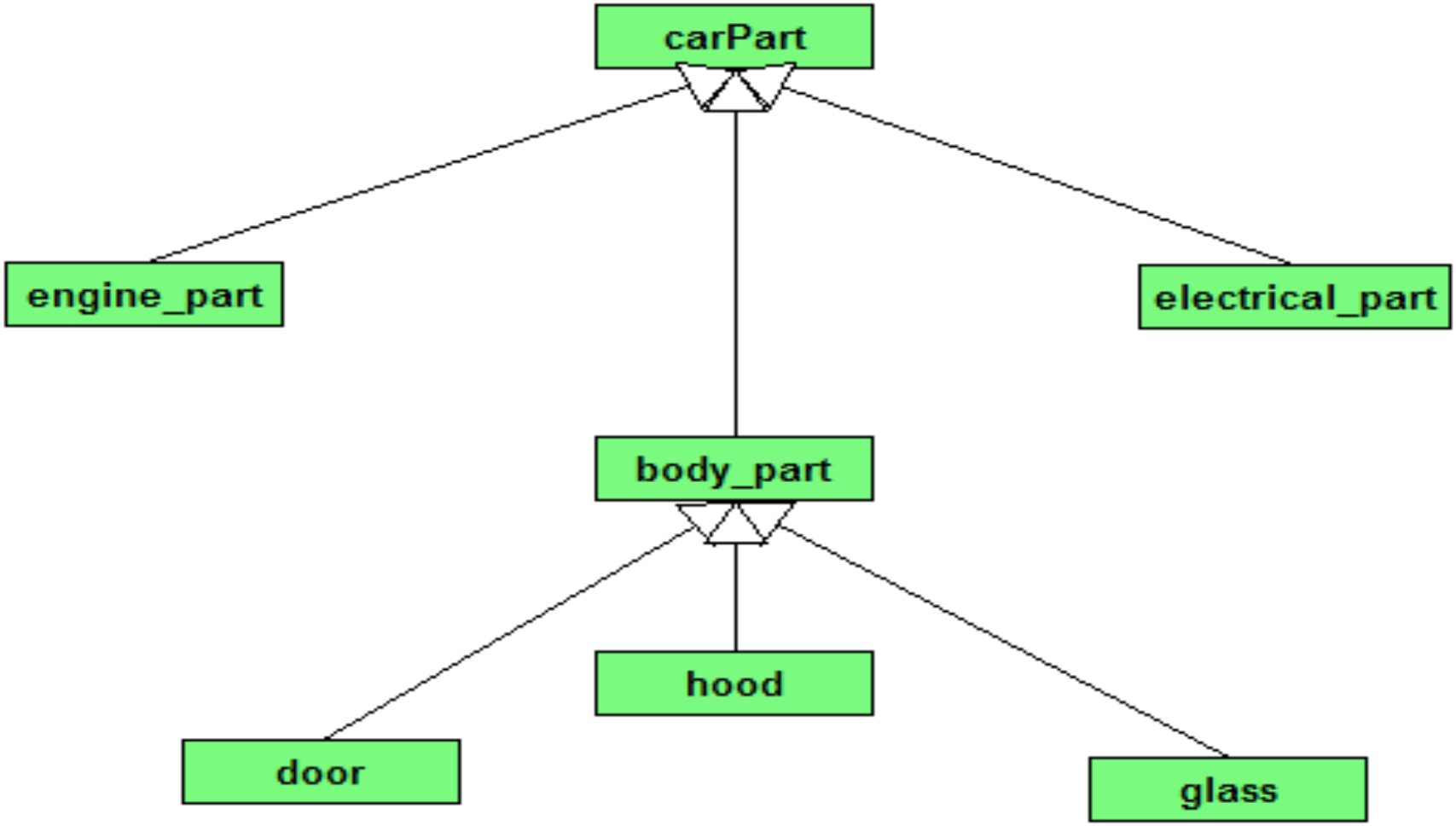




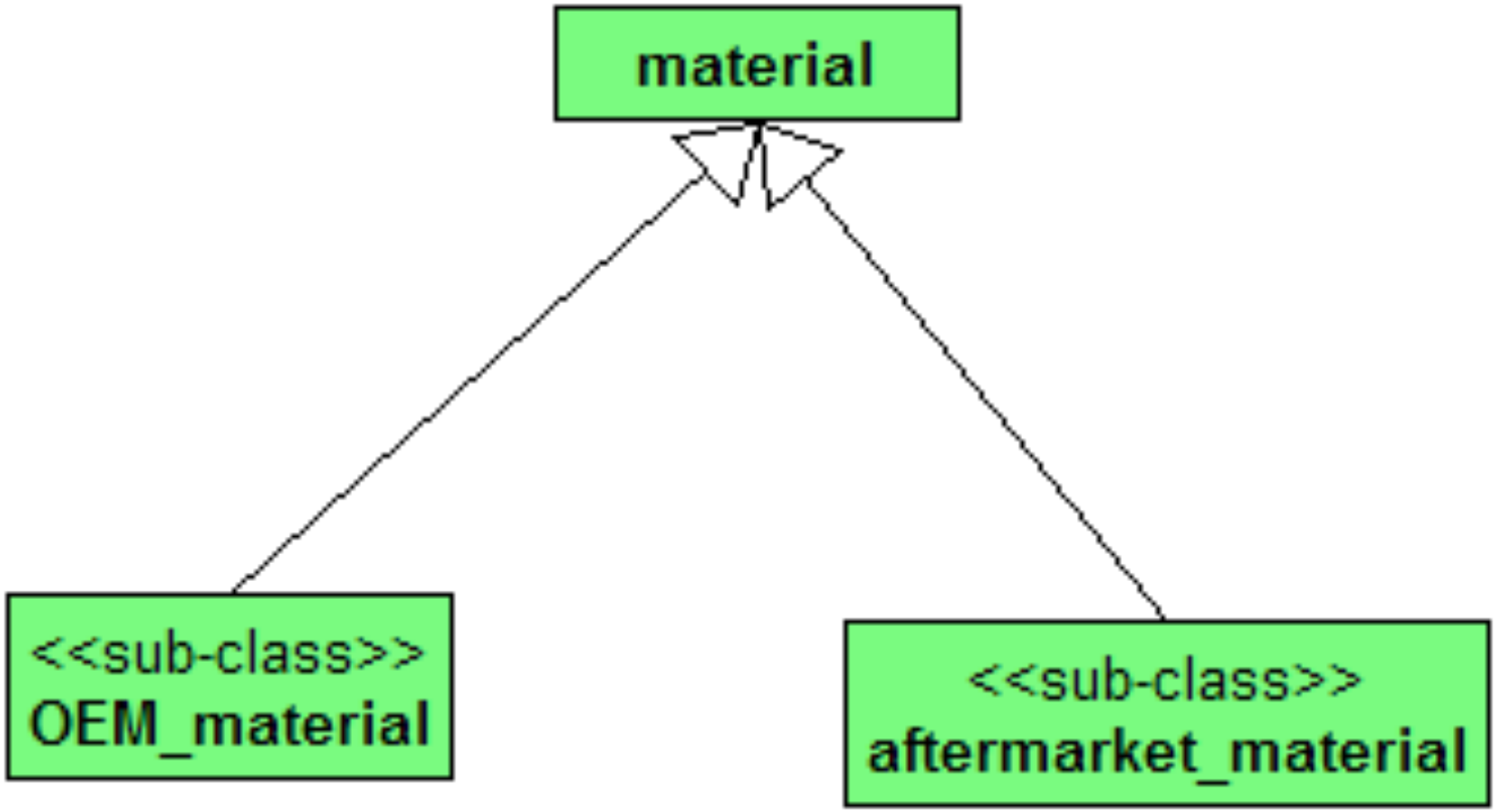
Ontology Languages

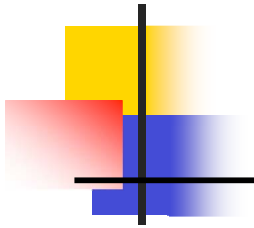
- If RDF can do all this, what more does an Ontology language offer ?
  - Relationships among resources
  - Relationships between properties
  - Cardinality, Ordinal values, constraints
  - Based upon RDF as a foundation

Ontology Model Example



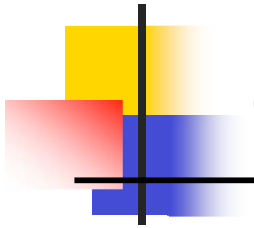
Properties Have Sub-properties





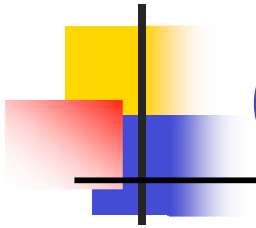
DAML + OIL

- Early Ontology languages
  - DAML
  - OIL
- The W3C Web Ontology working group started with both of the above and arrived at a standard – OWL



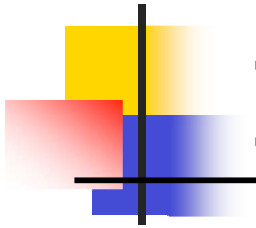
OWL

- OWL (Web Ontology Language)
- 3 flavors
  - OWL full – rich and complex
  - OWL DL – allows for verification of model
  - OWL lite – rarely used, RDF preferred



Operations on Models

- Import of ontology models
- Merging of models
- Deriving inferences based on the above

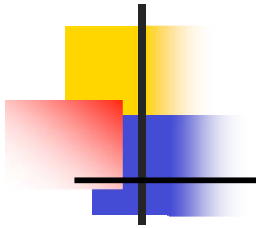


Inferences

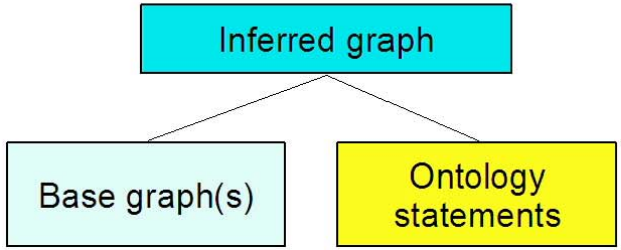
- Ontology + Reasoner = Inference

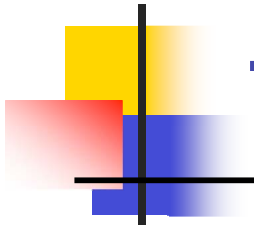
The result –

"***indirect statements that may not be a direct fact in the original RDF graph***"



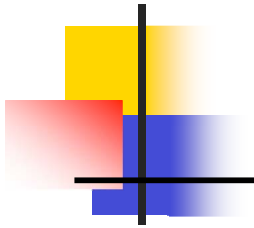
Inference Expands the Graph



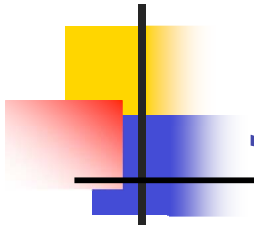


Tools I Like to Use

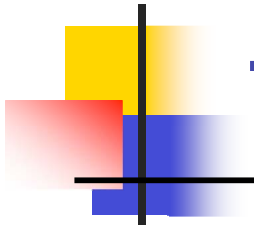
- RDF Author
- JENA
- Protégé



Protégé Walk Through

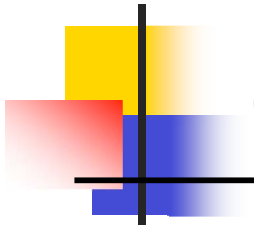


JENA Overview

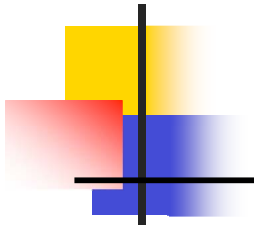


Tomorrow?

- True interoperability and extension of knowledge
  - Relationships and properties are stored in distinct repositories
  - At runtime these are brought together and knowledge is inferred from them
  - This differs from the common O-O paradigm where an object (or its class hierarchy) contains the information within itself

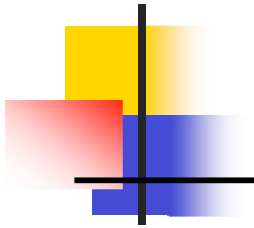


Questions?



Resources

- JENA download
  - http://jena.sourceforge.net
- Protégé web site
  - http://protege.stanford.edu



References

- http://www.w3.org/RDF/
- RDFGateway site
- Uche Obuji's articles on the IBM developerworks site – http://www.ibm.com/developerworks