The Semantic Web-on the respective Roles of XML and RDF

Stefan Decker et al
Presenter: Vijai P. Rao
What is this paper about?

- Give an introduction to the semantic web and the work that needs to be done over the RDF data model
- Sources.
  - w3.org
  - Ora Lassila’s slides
How we are going to go today

- What is the semantic web?
- What is an ontology?
- XML
- RDF
- Why RDF is better than XML
- RDF alone is not enough..
What is the Semantic Web?

- Aims at machine processable information
- This means..
- Machine's ability to solve a well-defined problem by performing well-defined operations on existing well-defined data (not AI!)
Why Semantic Web?

- To date, the World Wide Web has developed most rapidly as a medium of documents
- Information Overload
  - Information on the Web currently aiming at Human Consumption
- Search Engine fail more and more
  - Combined coverage is less than 42% of the HTML-Web
- Data interchange growing
  - Needs a common semantics
Why Semantic Web? Cont’d

- Make Web information practically processable by a computer
- Underlying this is the goal of making the Web more effective for its users (hotbot)
- Make a user’s life easier by the aggregation and creation of new, trusted information over the Web
Tim Berners-Lee’s Vision of Semantic Web
Relevance to Databases

- Similar problems to Distributed Databases

- Well..not similar in the sense that we aren't worrying about sharing rows or duplicating copies..infact we are concerning ourselves with machine processable information

- BUT, similar issues as in we need a master ontology, we have issues with scaling. As of now, we have to assume co-operation and trust since these layers haven't been implemented yet.
We need: Syntactic Interoperability

Enabling heterogeneous components to interact.

- Bridge mismatches in:
  - Data formats.
  - Language mechanisms
  - i.e parse any data
And..Semantic Interoperability

- Ability to agree on the *meaning* of data and operations.
An ontology:

An ‘Ontology’ is an agreed on, shared, common understanding of a domain written as an explicit, formal specification.
Ontologies

- It is clear that a semantics based web in general will be very useful.
- Yahoo, hotbot are search engines based on semantics (metadata, i.e label)
An Example Ontology

class-def animal
class-def plant
  subclass-of NOT animal
class-def defined carnivore
  subclass-of animal
  slot-constraint eats
    value-type animal
XML

- To store, carry and exchange data.
- Case sensitive, nesting must be exact and attributes must have quotes.
- The legal building blocks are defined by DTDs. A valid XML is one that is validated against a DTD.
- XML Schemas should soon replace DTDs: can extend restrict types to form subtypes.
Uses of XML

- XML can make your Data more Useful
- With XML, your data is available to more users.
- Since XML is independent of hardware, software and application, you can make your data available to other than only standard HTML browsers.
XML and Semantics

- XML Documents do not have semantics
- One uses XML to define an XML language adhering to a particular DTD
- XML documents can have semantics only by convention
- Implicit Semantic agreement on paper within a community of users for a particular domain data
- author> <uri>page</uri> <name>Ora</name> </author> or

- <document href="page"> <author>Ora</author> </document> or

- <document> <details> <uri>href="page"></uri> <author> <name>Ora</name> </author> </details> </document>
So..

Requires pre-arranged agreement

Only feasible for closed collaboration
  - agents in a small & stable community
  - pages on a small & stable intranet

not for sharable Web-resources
RDF

- RDF is a data model
  - the model can be viewed as directed, labeled graphs or as an object-oriented model (object/attribute/value)
  - the specification provides an encoding (in XML) of the model
  - RDF data model is an abstract, conceptual layer independent of XML
The RDF structure is based on the three main concepts: a resource, a property, and a statement.

- A resource is any web page that can be identified with the URI.
- A property is a resource with a name, such as Title.
- A statement is a resource, a property, and a value.
RDF Model Primitives

Resource  --Property-->  Value

Statement
Example

- Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila.
  - Subject(Resource): http://www.w3.org/Home/Lassila
  - Predicate (Property): Creator
  - Object (literal/value): "Ora Lassila"
More examples

<table>
<thead>
<tr>
<th>Resource (Subject)</th>
<th>Property (Predicate)</th>
<th>Value (Object)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;War and Peace&quot;</td>
<td>Author</td>
<td>&quot;Leo Tolstoy&quot;</td>
</tr>
<tr>
<td><a href="http://www.ccil.org/~cowan">http://www.ccil.org/~cowan</a></td>
<td>MIME Type</td>
<td>&quot;text/html&quot;</td>
</tr>
</tbody>
</table>
Also as OO

- RDF provides metadata about Web resources
- Object -> Attribute-> Value triples

Diagram:

- pers05
- Author-of
- ISBN...
Object -- Attribute -> Value triples

- objects are web-resources
- Value is again an object:
  - triples can be linked
  - data-model = graph
RDF example

RDF ~ set of (Resource, Property, Value)

"The Author of http://scom.hud.ac.uk/scomtlm/Artform/planning.html is Lee McCluskey."

IN RDF:

```xml
<rdf:Description about="http://scom.hud.ac.uk/scomtlm/Artform/planning.html">
  <Author>Lee McCluskey</Author>
</rdf:Description>
```

Resource, Property, Values can all have URI’s
We need an RDF Schema

- What *vocabulary* should we use ("Contains", "ChartType", etc)?
- Add traditional datatypes to RDF
  - the value of the property labeled should be a boolean *in the mathematical sense*
- Use RDF Schemas for the basic vocabulary
  - what properties can be used?
  - what resources the properties can be applied to?
So, RDF:
- (very small) commitment to modeling primitives
- but: no commitment to domain vocabulary

**RDF Schema**
- Define vocabulary for RDF
- Organize this vocabulary in a typed hierarchy
  - `Class`, `SubClassOf`, `type`
  - `Property`, `subPropertyOf`,
  - `domain`, `range`
### RDF and RDFS Properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Comment</th>
<th>domain</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs:isDefinedBy</td>
<td>Indicates the namespace of a resource</td>
<td>Resource</td>
<td>Resource</td>
</tr>
<tr>
<td>rdf:subject</td>
<td>The subject of an RDF statement.</td>
<td>Statement</td>
<td>Resource</td>
</tr>
<tr>
<td>rdf:predicate</td>
<td>the predicate of an RDF statement.</td>
<td>Statement</td>
<td>Property</td>
</tr>
<tr>
<td>rdf:object</td>
<td>The object of an RDF statement.</td>
<td>Statement</td>
<td>Not specified</td>
</tr>
<tr>
<td>rdf:type</td>
<td>Indicates membership of a class</td>
<td>Resource</td>
<td>Class</td>
</tr>
<tr>
<td>rdfs:member</td>
<td>a member of a container</td>
<td>Container</td>
<td>Not specified</td>
</tr>
<tr>
<td>rdfs:subClassOf</td>
<td>Indicates membership of a class</td>
<td>Class</td>
<td>Class</td>
</tr>
<tr>
<td>rdf:value</td>
<td>Identifies the principal value (usually a string) of a property when the property value is a structured resource</td>
<td>Resource</td>
<td>Not specified</td>
</tr>
<tr>
<td>rdfs:subPropertyOf</td>
<td>Indicates specialization of properties</td>
<td>Property</td>
<td>Property</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>Use this for descriptions</td>
<td>Resource</td>
<td>Literal</td>
</tr>
<tr>
<td>rdfs:label</td>
<td>Provides a human-readable version of a resource name.</td>
<td>Resource</td>
<td>Literal</td>
</tr>
<tr>
<td>rdfs:domain</td>
<td>A domain class for a property type</td>
<td>Property</td>
<td>Class</td>
</tr>
<tr>
<td>rdfs:range</td>
<td>A range class for a property type</td>
<td>Property</td>
<td>Class</td>
</tr>
<tr>
<td>rdfs:seeAlso</td>
<td>A resource that provides information about the subject resource</td>
<td>Resource</td>
<td>Resource</td>
</tr>
</tbody>
</table>
RDFS and RDF

- `<rdfs:Class rdf:ID="Teacher"> <rdfs:comment>Teacher Class</rdfs:comment> <rdfs:subClassOf rdf:resource="#Person"/> </rdfs:Class>

- `<teacher> <Teacher rdf:ID="jp"> <name>John Punin</name> </Teacher>

- Usually, schema link to
  - `http://www.w3.org/2000/01/rdf-schema#`
RDF & RDFS

- Provide a data model and syntax convention for representing the semantics of data in standardized interoperable manner
- Describe relationships among resources
- RDFS – Minimal ontology modeling language, object oriented type system
Comparitive XML Schema drawbacks

- XMLS can extend/restrict types to form subtypes but still no concept of inheritance.
- A union of possible types for an element is possible with the `<union>` tag, and this is not possible in RDF. However, RDF uses bags which allow for unordered collections. Moreover, the higher levels (like OWL) support unions.
Why RDF(S) Is Not Enough

- Expressive inadequacy
  - Only range/domain constraints (on properties)
  - No equivalence, disjointness, coverings etc.
  - No necessary and sufficient conditions (for class membership)
Extending RDF Schema

Goal
- make RDFS useable as ontology language
  - give RDF(S) precise semantics
  - extend RDF(S) with additional modeling primitives

Procedure
- formulate ontology language as RDF Schema document
  - using existing primitives as much as possible
  - placing additional primitives in the hierarchy of RDFS primitives
Ontology Inference Layer (OIL)

- Sponsored by European Union IST programme for Information Society Technologies
How DAML+OIL Builds ON RDFS

- Extends expressive power
  - Constraints (restrictions) on properties of classes (existential/universal/cardinality)
  - Boolean combinations of classes and restrictions
  - Equivalence, disjointness, coverings
  - Necessary and sufficient conditions
  - Constraints on properties
OIL (now DAML+OIL)

- Similar to RDFS in providing infrastructure to allow machines to make inferences
  Given
  
  (motherOf subProperty parentOf)
  (Mary motherOf Bill)

  when stated in OIL, allows you to conclude
  
  (Mary parentOf Bill)

- RDFS is limited in expressiveness that OIL addresses
How does OIL translate to RDF

- Simply map OIL to RDF vocabulary

<table>
<thead>
<tr>
<th>OIL</th>
<th>RDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class-def</td>
<td>rdfs:Class</td>
</tr>
<tr>
<td>Subclass-of</td>
<td>rdfs:Subclass</td>
</tr>
<tr>
<td>Slot constraint</td>
<td>Subclass expressions in RDF-OIL</td>
</tr>
<tr>
<td>AND, “,”</td>
<td>oil:AND</td>
</tr>
<tr>
<td>Has – value</td>
<td>Oil:has - value</td>
</tr>
</tbody>
</table>
• class-def
• subclass-of
• slot-def
• subslot-of
• domain
• range
• class-expressions
  • AND, OR, NOT
• slot-constraints
  • has-value, value-type
  • cardinality
• slot-properties
  • trans, symm
## Language Feature Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>XML DTD</th>
<th>XML Schema</th>
<th>RDF(S)</th>
<th>DAML+OIL</th>
<th>RDF(S) 2002</th>
</tr>
</thead>
<tbody>
<tr>
<td>bounded lists</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>cardinality constraints</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>class expressions</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>data types</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>?</td>
</tr>
<tr>
<td>defined classes</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>enumerations</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>equivalence</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>extensibility</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>formal semantics</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>inheritance</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>inference</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>local restrictions</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>qualified constraints</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>reification</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
What did we get?

- Any RDF agent can process OIL instances
- Any RDF-S agent can process OIL ontologies
- Any OIL-aware agent can exploit semantics & reasoning
Conclusion..proof and trust!?

- **Logic**
  - I am an employee of UMBC.
  - UMBC is a member of W3C.
  - UMBC has GET access to http://www.w3.org/Member/.
  - I (therefore) have access to http://www.w3.org/Member/.

- **Proof**
  - UMBC's document employList lists me as an employee.
  - W3C's member list includes UMBC.
  - The ACLs for http://www.w3.org/Member/ assert that employees of members have GET access.

- **Trust**
  - UMBC's document employList is signed by a private key that W3C trusts to make such assertions.
  - W3C's member list is trusted by the access control mechanism.
  - The ACLs for http://www.w3.org/Member/ were set by an agent trusted by the access control mechanism.
Latest..OWL=OIL+DAML

- OWL adds more vocabulary for describing properties and classes: relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes.
Comments:

- Remember that the paper isn’t saying RDF is better than XML. It is just saying that RDF is better for the semantic web.
Conclusion

- Meta-data should be given lots of importance in the semantic web
- Work needs to be done on establishing trust and security.