RDF (Resource Description Framework)

1. RDF provides a way of describing resources via metadata (data about data) and it restricts the description of resources to triplets (subject, predicate, object).
2. It provides interoperability between applications that exchange machine understandable information on the Web.
3. The broad goal of RDF is to define a mechanism for describing resources that makes no assumptions about a particular application domain, nor defines (a priori) the semantics of any application domain.
   - Uses XML as the interchange syntax.
   - Provides a lightweight ontology system.

The formal specification of RDF is available at:
http://www.w3.org/TR/REC-rdf-syntax/

RDF Syntax

Subject, Predicate and Object Triplets (Tuples)

- Subject: The resource being described.
- Predicate: A property of the resource
- Object: The value of the predicate

A combination of them is said to be a Statement (or a rule)


A web page being described  A property of the web page (author)

[Subject]  [Predicate]  [Object]

RDF Example

```xml
<html 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/"
    xmlns:dct="http://purl.org/dc/terms/"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:xmlns="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
>
  <rdf:Description about="http://foo.bar.org/index.html">
    <dc:title>Example</dc:title>
    <dc:creator>John Doe</dc:creator>
  </rdf:Description>

  <!-- Example (cont..) -->

</html>
```

RDF Schema

A schema defines the terms that will be used in the RDF statements and gives specific meanings to them.

http://www.w3.org/TR/rdf-schema/

RDF: Tools/Resources

- SipPAC: A Simple RDF Parser & Compiler. It parses the RDF, and validates it. It also generates the tuples and even draws a graph of the data model.
  - Website: www.w3.org/RDF/Implementations/SiRPAC/
- Protege: A Nice Metadata Editor. Java based simple user interface to describe a web resource. Can mail the metadata file to yourself after finished editing.
  - Website: http://metadatlab.net/ Protege/
- Reggie: Editor of ontologies in practically any language you care about. Open source.
  - Website: http://www.smi.stanford.edu/projects/protege/
Summary: RDF & RDF Schema layer

- Minimalist model - (thing), Class, Property
- Subproperty, Subclass
- Domain & Range

- Still not a W3C recommendation
- Continues to change
- Other languages are being built on XML substrate: XQUERY, XTM

The Layer Cake [TBL,XML2000]

Limitations of RDF

- Cannot define properties of properties (unique, transitive)
- No equivalence, disjointness, etc.
- No mechanism of specifying necessary and sufficient conditions for class membership.

  Example:
  - If it is given that 'XYZ' has a 'car' which is '7ft high', has 'wide wheels' and 'loading space is 4 cub.m', then we should be able to reason that 'XYZ' has an 'SUV', as given by the necessary and sufficient conditions for being an 'SUV': height > 4ft & wide wheels & loading space > 2 cub.m

DAML+OIL’s History

- W3C’s Semantic Web Activity:
  - RDF and metadata markup efforts to represent data in a machine understandable form.

- DARPA started the DARPA Agent Markup Language (DAML) program.
  - possibly with “ARPANET -> Internet” in mind

- EC (European Commission) funding programs
  - Ontology Interchange Language (OIL)
  - logic based language.
  - brings logic and inference to the Semantic Web

www.daml.org

DAML+OIL: http://www.daml.org/2001/03/daml+oil-index.html

DAML+OIL (www.daml.org)

- It builds on earlier W3C standards such as RDF and RDF Schema.
- DAML extends RDF and RDFS with richer modelling primitives.
  - disjointWith, intersectionOf, oneOf, cardinality
- Able to provide properties of properties
  - uniqueness, transitivity, etc.
- Current version DAML+OIL provides a semantic interpretation (model-theoretic semantics)

http://www.daml.org/2001/03/daml+oil-index.html

An Example (from www.daml.org)

```xml
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:daml="http://www.daml.org/2000/12/daml+oil#"
  xmlns="http://www.daml.org/2000/12/daml+oil-ex#">
  <daml:Ontology about="">
    <daml:versionInfo>An example ontology</daml:versionInfo>
    <rdfs:Class rdf:ID="Animal">
      <rdfs:label>Animal</rdfs:label>
      <rdfs:comment>This class of animals is illustrative of a number of ontological idioms.</rdfs:comment>
    </rdfs:Class>
    <rdfs:Class rdf:ID="Male">
      <rdfs:subClassOf rdf:resource="#Animal"/>
    </rdfs:Class>
    <rdfs:Class rdf:ID="Female">
      <rdfs:subClassOf rdf:resource="#Animal"/>
      <daml:disjointWith rdf:resource="#Male"/>
    </rdfs:Class>
    <rdfs:Class rdf:ID="Man">
      <rdfs:subClassOf rdf:resource="#Person"/>
      <rdfs:subClassOf rdf:resource="#Male"/>
    </rdfs:Class>
  </daml:Ontology>
</rdf:RDF>
```

Can explicitly specify the set of Females to be disjoint with the set of Males

The Person class is defined later

To be read conjunctively. A man is a sub-class of 'Person' and a 'Male'
Example (contd..)

```
<rdfs:Class rdf:ID="Woman">
  <rdfs:subClassOf rdf:resource="#Person"/>
  <rdfs:subClassOf rdf:resource="#Female"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#hasParent"/>
      <daml:toClass rdf:resource="#Person"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#hasFather"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</rdfs:Class>
```

Restrictions on the property hasParent (only for the Person class – Local scope, as opposed to rdfs:range)

A person can have only another Person as its parent

Further constructs that the example doesn’t use:

- Properties: TransitiveProperty (hasAncestor), UniqueProperty (hasMother), inverseOf(hasChild -> hasParent), etc.
- Classes: intersectionOf (a daml:collection), unionOf (a daml:collection), sameClassAs, complementOf, etc.

DAML References/Tools

- DAML Viewer: http://www.daml.org/viewer/applet.html
- DAML Crawler Results: http://www.daml.org/crawler/pages.html
- A DAML Validator: http://www.daml.org/validator/
- A DAML example explained: http://www.daml.org/2001/03/daml-oil-walkthru.html

The Layer Cake [TBL,XML2000]

W3C’s Semantic Web Principles

1. Everything identifiable is in the Semantic Web (URIs!)
2. Partial information
   - Anyone can say anything about anything
3. Web of trust
   - All statements on the Web occur in some context
4. Evolution
   - Allow combining independent work done by different communities
5. Minimalist design
   - Make the simple things simple, and the complex things possible
   - Standardize no more than is necessary
Hypertext: Then and Now

- SOTA circa 1990: Dynatext’s electronic book
  - A book had to be compiled (like a program) in order to be displayed efficiently
  - A central link database, to make sure there were no broken links
  - Text that was fixed and consistent (a whole book)
- WWW:
  - Links can be added and used at any time
  - Distributed (must live with broken links!)
  - Decentralized

Knowledge Representation: Now and Tomorrow

“To webize KR in general is, in many ways, the same as to webize hypertext. Replace identifiers with URIs. Remove any requirement for global consistency. Put any significant effort into getting critical mass. Sit back.”

-- TBL

Ongoing Work at ISI

- EXPECT (knowledge acquisition and problem solving)
  - No longer developing KBs, but importing schemas and data
- Electric Elves
  - Agents are more transparent and publish data & schemas, advertisements/assumptions
- TRELLIS (try it out at trellis.semanticweb.org!)
  - Users represent decisions and opinions -> Web of Trust
- IKRAFT
  - Users turn text in progressively more formal representations (KB) -> semi-formal annotations