

RDF, OWL, and Protégé

By:

A.Aziz Altowayan @ Pace University

3/25/2014

Agenda

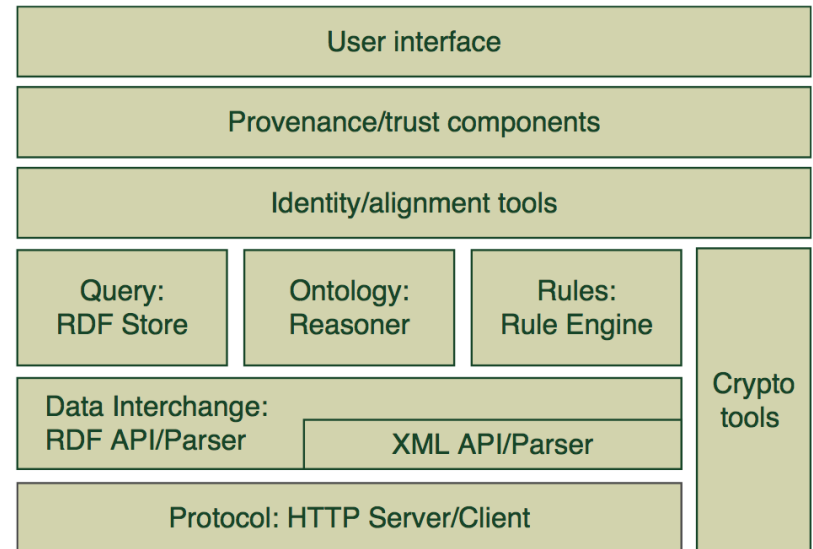
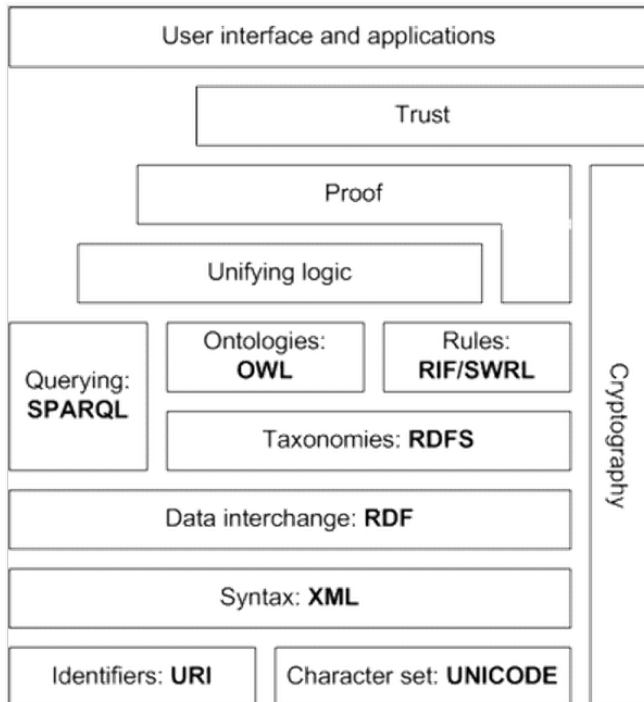
- **Semantic Web Architecture**
- **RDF**
- **OWL**
- **Using Protégé**

Semantic Web Architecture

SemWeb architecture is illustrated as a hierarchy (Stack or layers) of languages used to create semantic web (as of Berners-Lee, 2000).

Each layer exploits and use the capabilities of the layers below.

- Hypertext technologies:
 - Unicode, URI, and XML
- SemWeb technologies:
 - RDF, RDFS, OWL, SPARQL, and RIF.
- Unrealized SemWeb Technologies:
 - Cryptography, Trust, and UI.



See:

- Ch.2, Handbook of SemWeb Technologies
- [semweb-Arch-wiki](#)
- [SemWeb Arch.](#)

Separation of concepts (meanings)

XML: a markup language for structuring exchangeable data.

RDF: a general framework for modeling web resources information.

OWL: a language for authoring (creating) web ontologies.

Ontology: a set of concepts representing a domain of knowledge.

“RDF: makes statements.

OWL: makes vocabulary.”

Dr. Tao

Why RDF and not XML?

RDF vs. XML:

- RDF uses XML syntax and its namespace concept.
- RDF is a more restricted form of XML.
- XML cannot accomplish what RDF does, why:
 - a. XML provides limited and ambiguous semantic.
 - b. Parsing XML statements is not scalable (depends on tree structure).
 - c. RDF promotes development and usage of standardized ontologies.

Two reasons for RDF:

- Information maps directly to a model (implies the semantic).
- RDF data can be part of Semantic Web.

What is RDF

- RDF is a data-format or framework for linked data.
- RDF uses *triples*: **subject-predicate-object** (aka subject-property-value).
- Triples are used to describe relationships (link) between web resources.
- Resources can be anything (a webpage, a person, or a thing ... etc) represented as URIs ([IRIs](#)).

Representing info (resources) using RDF

RDF Triples can be expressed [using either](#):

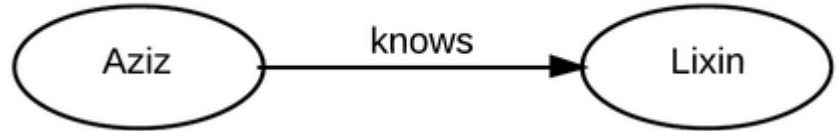
- English, Graph (represented textually in Turtle), or XML-like syntax.

Example: Aziz knows Lixin.

```
uri://people#Aziz
```

```
http://xmlns.com/foaf/0.1/knows
```

```
uri://people#Lixin
```



RDF is mainly about connecting (linking) resources “by making statements”.

What is OWL

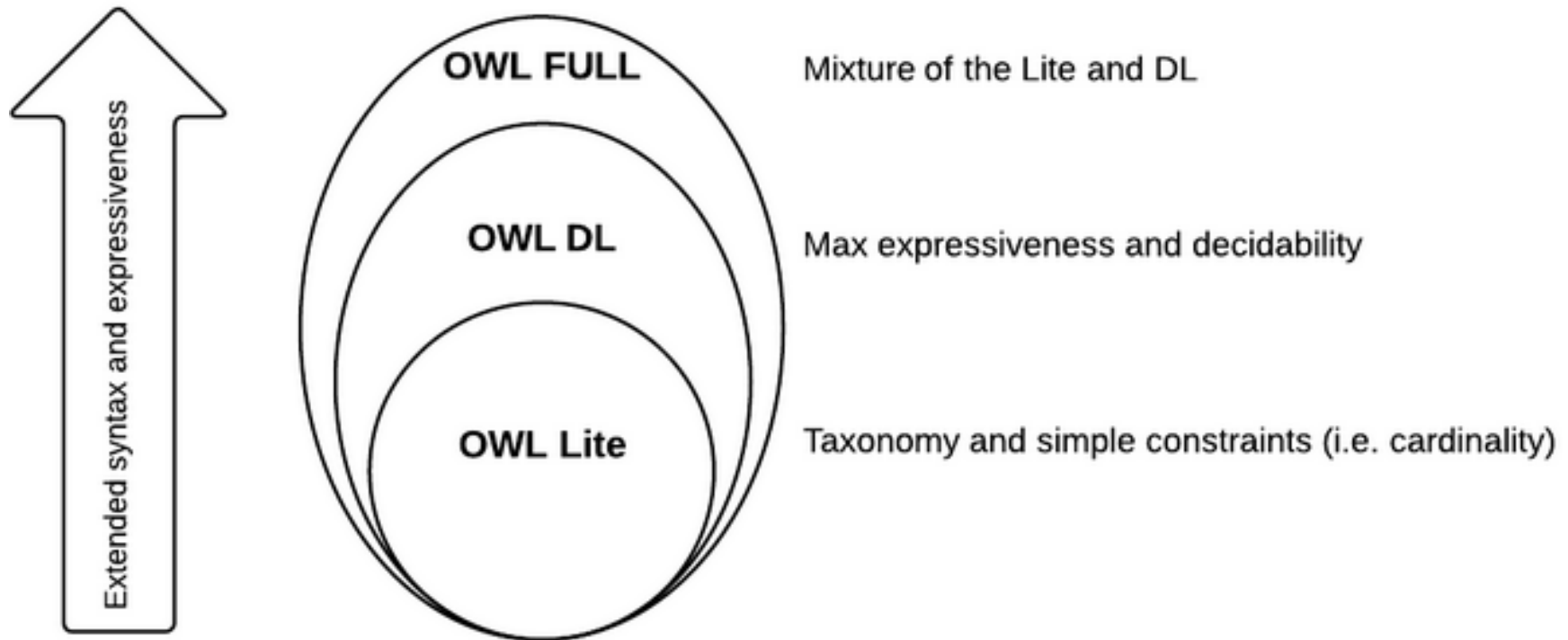
- OWL is an extension of RDFS for using **Description Logic (DL)**.
- Thus, with **description logic**, it brings **reasoning power** to SemWeb.
 - with a set of operators such as: intersection, union, and negation.

OWL is all about using Description Logic.

Reasoning is deriving facts that are not expressed in ontology or in knowledge base explicitly.

OWL sublanguages

OWL variants, and their level of expressiveness



See:

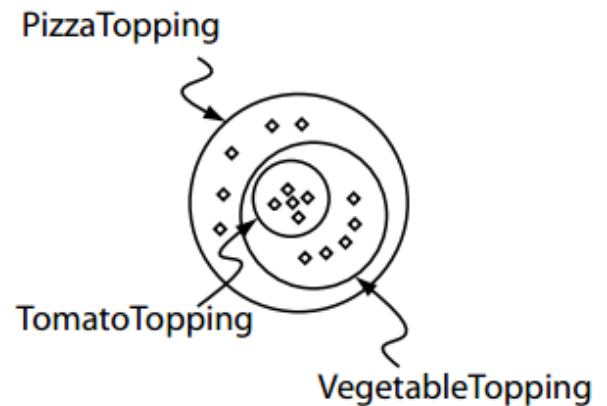
[OWL DL Semantics](#)

Using Protégé to create ontology

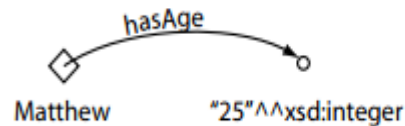
OWL consists of: **individuals**, **properties**, and **classes**.

In Protégé, these corresponds to: **instances**, **slots**, and **classes**.

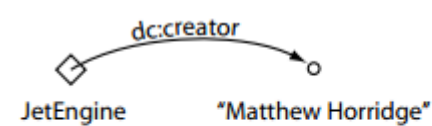
- **Classes (subclasses and disjoint).**
- **Properties**
 - *Object property*: relationship between two individuals.
 - *Datatype property*: link individual to an XML Schema Datatype value or an RDF literal.



An object property linking the individual Matthew to the individual Gemma



A datatype property linking the individual Matthew to the data literal '25', which has a type of an xsd:integer.



An annotation property, linking the class 'JetEngine' to the data literal (string) "Matthew Horridge".

See:

[Protégé tutorial](#)

Protégé Demo with Camera

The screenshot displays the Protégé web interface for the 'Camera' ontology. The browser address bar shows the URL: <http://www.semanticweb.org/aziz/practice-onto/Camera>. The interface includes a search bar for entities and a navigation menu with tabs for Active Ontology, Entities, Classes, Object Properties, Data Properties, Annotation Properties, Individuals, OWLViz, DL Query, OntoGraf, Ontology Differences, and SPARQL Query.

The main content area is divided into two panels. The left panel, titled 'Class hierarchy: SLR', shows a tree view of the ontology. The 'SLR' class is highlighted in orange. The hierarchy is as follows:

- Thing
 - Camera
 - Digital ≡ DigitalCamera
 - PointAndShoot
 - SLR ≡ SingleLensReflex**
 - Film
 - datatype
 - decimal
 - MegaPixel
 - DigitalCamera ≡ Digital
 - Person
 - Photographer
 - Amateur
 - Professional
 - SingleLensReflex ≡ SLR
 - specifications

The right panel, titled 'Annotations: SLR', shows the class annotations for SLR, which are currently empty. Below this is the 'Description: SLR' panel, which lists the following relationships:

- Equivalent To: **SingleLensReflex**
- SubClass Of: **Digital**
- SubClass Of (Anonymous Ancestor): **SLR**, **DigitalCamera**, **Digital**
- Members: (empty)
- Target for Key: (empty)
- Disjoint With: **PointAndShoot**

The bottom left panel, titled 'Object property hierarchy', shows a list of object properties:

- topObjectProperty
 - betterQualityPriceRatio**
 - friend_with**
 - has_spec**
 - own**
 - owned_by**
 - pixel**

Protégé Demo with Camera

