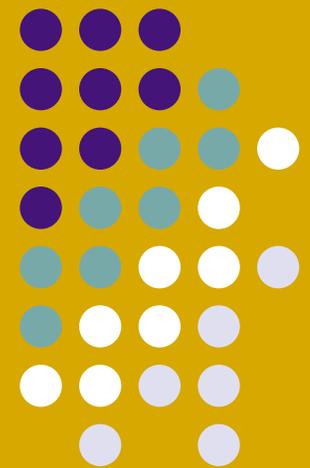


# Semantic Web

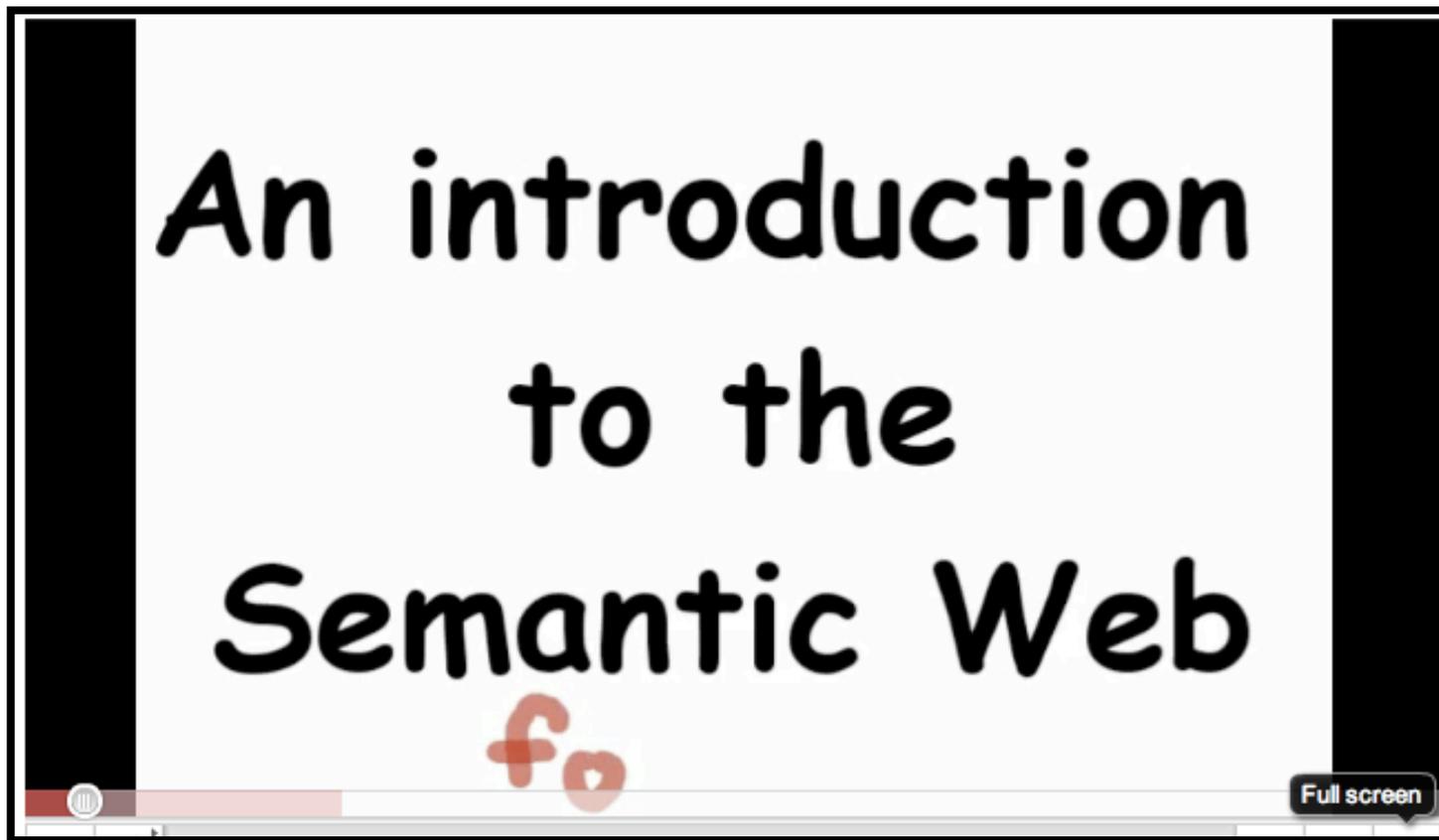


Dongwon Lee, Ph.D.  
(Many Examples by A.M. Kuchling)



# Semantic Web Video

<http://youtu.be/OGg8A2zfWKg>



# 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, James Hendler, Ora Lassila, [The Semantic Web](#), Scientific American, May 2001]*



# Motivation

- Current World-Wide Web is a geomantic conglomeration of information in distributed and heterogeneous environment
- Yet, the processing of such information is merely based on the **syntactic** matching of information
  - Eg, Web search using keyword matching
- More intelligent processing is needed with requirements:
  - **Machine processing**
  - **Automatic processing**

# Current Web: Syntactic Web

KOREA UNIVERSITY BUSINESS SCHOOL

Home > Faculty & Research > Faculty > MIS

Faculty - MIS

Name: **Dongwon Lee** Position: Associate Professor

Tel: 82-2-3290-2822 E-mail: mislee@korea.ac.kr

Ph.D. Degree: University of Minnesota

Research Areas

Teaching & Research

→ HOMEPAGE

PENNSSTATE

Overview

Schedule

Research

Teaching

Services

Miscellaneous

**Dongwon Lee**

Ph.D., UCLA, 2002  
Associate Professor  
College of Information Sciences and Technology (IST)  
Penn State University

Two prof “Dongwon Lee”:  
One at PSU and  
the other at Korea U.



I am an associate professor of [College of Information Sciences and Technology \(IST\)](#) at [Penn State University](#). In 2002, I received my Ph.D from [Computer Science Department](#) of [UCLA](#) with the [dissertation](#) on the [Database](#) and [XML](#).

My not-so-fancy CV is [here](#), and my PGP public key is [here](#).

Email : [dongwon@psu.edu](mailto:dongwon@psu.edu)

Phone : 814.865.0687

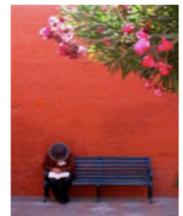
Fax : 814.865.6426

313A IST Building

Address : Penn State University

University Park, PA 16802-6823

Assistant Staff : Susan V. Vactor (814.865.6453)



# Current Web: Syntactic Web

## Van, Turkey

From Wikipedia, the free encyclopedia

Coordinates: 38°29′39″N 43°22′48″E﻿ / ﻿38.49417°N 43.37999°E﻿ / 38.49417; 43.37999

*This article is about a city in Turkey. For other uses, see [Van \(disambiguation\)](#).*

**Van** (Armenian: Վան<sup>[1]</sup>) is a city in eastern Turkey and the seat of Van Province, and is located on the eastern shore of Lake Van. The city's official population in 2009 was 360,810,<sup>[2]</sup> but many estimates put this as much higher with a 1996 estimate stating 500,000<sup>[3]</sup> and former Mayor Burhan Yengun quoted as estimating the population at approximately 600,000<sup>[4]</sup>

### Contents [hide]

- History
  - Urartu
  - From the Orontids to the Kingdom of Armenia
  - The Byzantines and the kingdom of Vaspurakan
  - The Seljuk Empire
  - The Ottoman Empire
    - City life
    - Demographics
  - World War I and the Armenian Genocide
  - Turkish War of Independence and Republic
- Van today
- Cuisine

### Van

— City —



Cars > Vans

Related Searches

#### Used Vans for Sale



#### 2004 Ford Econoline 350 for \$12,775



2004 Ford Econoline, 83,384 miles, White

Color-keyed engine console cover w/dual stowage, (4) cup holders, Black plastic stepwell pads,...

Pro

Posted by **Haldeman Ford** with **Oodle Pro**

Hamilton, NJ · 10 hours ago



#### 2009 Chevrolet Express 2500 WORK VAN for \$15,619



2009 Chevrolet Express, 34,127 miles, White

Audio system, radio provisions only. Includes alarm warning chimes, radio wiring harness, radio...

Pro

Posted by **Archer Kia Volkswagen Automotive** with **Oodle Pro**

# Current Web: Syntactic Web

Q: Search Chevrolet Express Van with less than 10K miles  
→ Currently, hard to answer this query !

Cars > Vans

Related Searches

Used Vans for Sale



Hamilton, NJ · 10 hours ago

## 2004 Ford Econoline 350 for \$12,775



2004 Ford Econoline, 83,384 miles, White

Color-keyed engine console cover w/dual stowage, (4) cup holders, Black plastic stepwell pads,...



Posted by **Haldeman Ford** with **Oodle Pro**



## 2009 Chevrolet Express 2500 WORK VAN for \$15,619



2009 Chevrolet Express, 34,127 miles, White

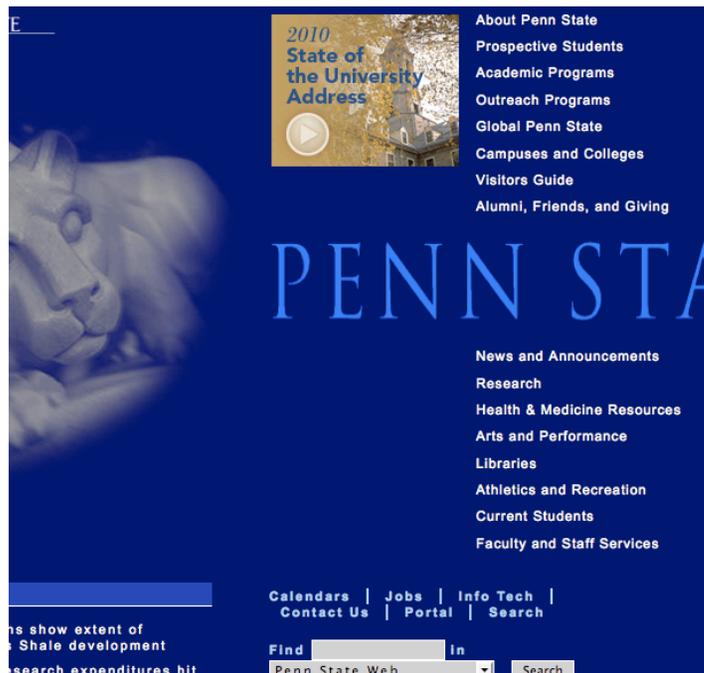
Audio system, radio provisions only. Includes alarm warning chimes, radio wiring harness, radio...



Posted by **Archer Kia Volkswagen Automotive** with **Oodle Pro**



# Current Web: Syntactic Web



- Markup consists of:
  - Rendering information (e.g., font size and colour)
  - Hyper-links to related content
- Semantic content is accessible to humans but not (easily) to computers...

# What Human Sees



About Penn State  
Prospective Students  
Academic Programs  
Outreach Programs  
Global Penn State  
Campuses and  
Colleges  
Visitors Guide  
Alumni, Friends, and  
Giving ...

# What Machine Sees

⊕ ⊕ ⊕ 📄 📁 📁 📄

❄️ ⚡️ ♁ ♁ ● ♁ ✦ ♁ ■ ◆ ⚡️

⌘ ■ ◆ ♁ □ ■ ☹️ ◆ ⌘ □ ■ ☹️ ● ◆ □ □ ● Ω

◆ ⌘ Ω ♁ ◆ ♁ Ⓞ ♁ □ ■ ✈️ ♁ □ ♁ ■ ♁ ♁

💧 ⚡️ ♁ □ ☹️ ◆ □ ■ ◆ ☹️ ⌘ & ⌘ & ⌘  
⚡️ □ ◆ ♁ ●

👉 □ ■ □ ● ◆ ● ◆ 📁 ⚡️ ☹️ ◆ ☹️ ⌘ ⌘ 📁 ✕ 💧 📄

🖨️ 📁 📁 📁 📁 ○ ☹️ ⏏️ 📄 📁 📁 📄

📁 ● □ ♁ ☹️ ◆ ⌘ □ ■ 📄 Ω ☹️ ⏏️ ◆

● ♁ ☹️ □ ■ ⌘ ■ ◆ ♁ □ ☹️ ♁ ◆

☀️ ♁ 📄 ⌘ ◆ ◆ ♁ □ ♁ Ω

□ ☹️ □ ◆ ⌘ ♁ ⌘ □ ☹️ ■ ◆ ◆ ♁ □ ○ ⌘ ■ 📄

# Solution #1: XML w. Meaningful Tags

<School>

<Name>⌘⌘⌘📄📁📁📄</Name>

<TOC>❄️⚡️📄 📄●📄❖📄■◆⚡️  
⌘■◆📄□■🌀◆⌘□■🌀● ◆□□●⊖ ◆⌘⊖📄  
◆📄⊖ 📄□■♊️📄□📄■📄📄</TOC>

<Description>💧⚡️📄□🌀◆□■ ◆🌀⌘&⌘&⌘  
⚡️□◆📄●

👉□■□●◆●◆📁 ⚡️🌀◆🌀⌘⌘📁 ✝️💧✌️

🖨️📁📁📁 ○🌀📁 📄📁📁📄

📁 ●□📄🌀◆⌘□■ 📄 ⊖🌀📁◆ ●📄🌀□■  
⌘■◆📄□🌀📄◆</Description>

</School>





# What is Needed? → Semantic Web

- **External agreement** on meaning of annotations
  - E.g., Dublin Core (DC) agree on the meaning of a set of annotation tags for “documents”
  - But, limited number of things can be expressed
- Use **Ontologies** to specify **meaning** of annotations
  - Ontologies provide a vocabulary of terms
  - New terms can be formed by combining existing ones
  - Meaning (i.e., **semantics**) of such terms is formally specified
  - Can also specify **relationships** between terms in multiple ontologies

# Solution #2: XML w. Semantic Tags

<School>

<Name>⊕⊕⊕📄📁📁📄</Name>

<TOC>❄️⚡️📄 📄●📄❖📄■◆⚡️  
⊕■◆📄□■🌀◆⊕□■🌀● ◆□□●⊖ ◆⊕⊖📄  
◆📄🌀 📄□■♊️📄□📄■📄📄</TOC>

<Description>💧⚡️📄□🌀◆□■ ◆🌀⊕&⊕&⊕  
⚡️□◆📄●

👉□■□●◆●◆📁 ⚡️🌀◆🌀⊕⊕📁 ✕💧✌️

🖨️📁📁📁📁 ○🌀📁📁 📄📁📁📄

📁 ●□📄🌀◆⊕□■ 📄 ⊖🌀📁◆ ●📄🌀□■  
⊕■◆📄□🌀📄◆</Description>

</School>



# What Machine Sees from Solution #2

Official title of an institution

<School>

<Name> ☉ ☉ ☉ 📄 📁 📁 📄 </Name>

<TOC> ❄️ ⚡️ ♀️ ♀️ ● ♀️ ✨ ♀️ 📁 ▲ ⚡️  
) ( 📁 ♀️ 📁 📁 ☉ ♀️ 📁 📁 ☉  
◆ ♀️ ☽ ♀️ 📁 📁 ✂️ ♀️ 📁 📁 ♀️ 📁 ♀️ ♀️ ♀️ </TOC>

Same meaning as <desc> from  
The namespace: <http://foo.com/myown>

<Description> ☉ ⚡️ ♀️ 📁 ☉ ◆ 📁 📁 ◆ ☉ ♂️ ♂️ ♂️  
⚡️ 📁 ◆ ♀️ ●

👉 📁 📁 📁 ● ◆ ● ◆ 📁 ⚡️ ☉ ◆ ☉ ♂️ ♂️ 📁 ☉ 📁 📁 📁  
📁 ● 📁 ♀️ ☉ ◆ ♂️ 📁 📁 📁 ☉ ☉ ☉ 📁 ◆ ● ♀️ ☉ 📁 📁  
) ( 📁 ♀️ 📁 ☉ ♀️ ◆ </Description>

</School>

Clear and precise meanings  
of tags known to machines

# W3C's Ontology Languages

- RDF (Resource Description Framework)
  - Graphical formalism to represent metadata and describe semantics
  - <http://www.w3.org/RDF/>
- RDF-S (RDF Schema)
  - RDF with schema vocabulary
  - Class, Property, type, subclassOf, subPropertyOf, range, domain
  - <http://www.w3.org/TR/rdf-schema/>
- OWL (Web Ontology Language)
  - Knowledge representation language for making ontologies
  - Superset of RDF and RDF-S

# RDF vs. RDF Schema vs. OWL

- RDF: **assertion of facts**
  - Resource X is named “Lee”
  - “Lee” is the author of resource X
- RDF Schema: **vocabulary definition**
  - There is a class called “Person”
  - “Dongwon Lee” is an instance of "Person"
- OWL: **relationship among vocabularies**
  - "Persons" in vocabulary A are the same thing as "People" in vocabulary B
  - Resource X and resource Y are referring to the same thing
  - If  $X \rightarrow Y$  and  $Y \rightarrow Z$ , then  $X \rightarrow Z$

# #1. RDF: Basic Concepts

- RDF is a language for representing information about **resources** in the Web
  - <http://www.w3.org/TR/rdf-primer/>
- RDF is particularly intended for representing **metadata** about Web resources, such as
  - Title, author, and modification date of a Web page
  - Copyright and licensing information about a Web document,
  - Availability schedule for some shared resource, etc
- RDF is intended for situations in which this information needs to be processed by **software applications**, rather than being only displayed to people

# Graph Data Model: “Triples”

- The underlying structure of any expression in RDF is a collection of triples
- Each consists of a subject, a predicate and an object
  - Subject
  - Object (or literal)
  - Predicate that denotes a relationship
- Triple notation: <subject, predicate, object>



# Triple Example 1

- Imagine trying to state that someone named “John Smith” created a particular Web page.
  - <http://www.example.org/index.html> has a **creator** whose value is **John Smith**
- **Subject**: the thing the statement describes (the Web page)
- **Object**: the thing the statement says is the value of this property (who the creator is), for the thing the statement describes
- **Predicate**: a specific property (creator) of the thing the statement describes

# Triple Example 1

<http://www.example.org/index.html>

<http://purl.org/dc/elements/1.1/creator>

<http://www.example.org/staffid/85740>

[http://en.wikipedia.org/wiki/Resource\\_Description\\_Framework](http://en.wikipedia.org/wiki/Resource_Description_Framework)

# Triple Example 1

```
<?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.example.org/index.html">
```

```
  <dc:creator rdf:resource = "http://www.example.org/staffid/85740"/>
```

```
</rdf:Description>
```

```
</rdf:RDF> ♪
```



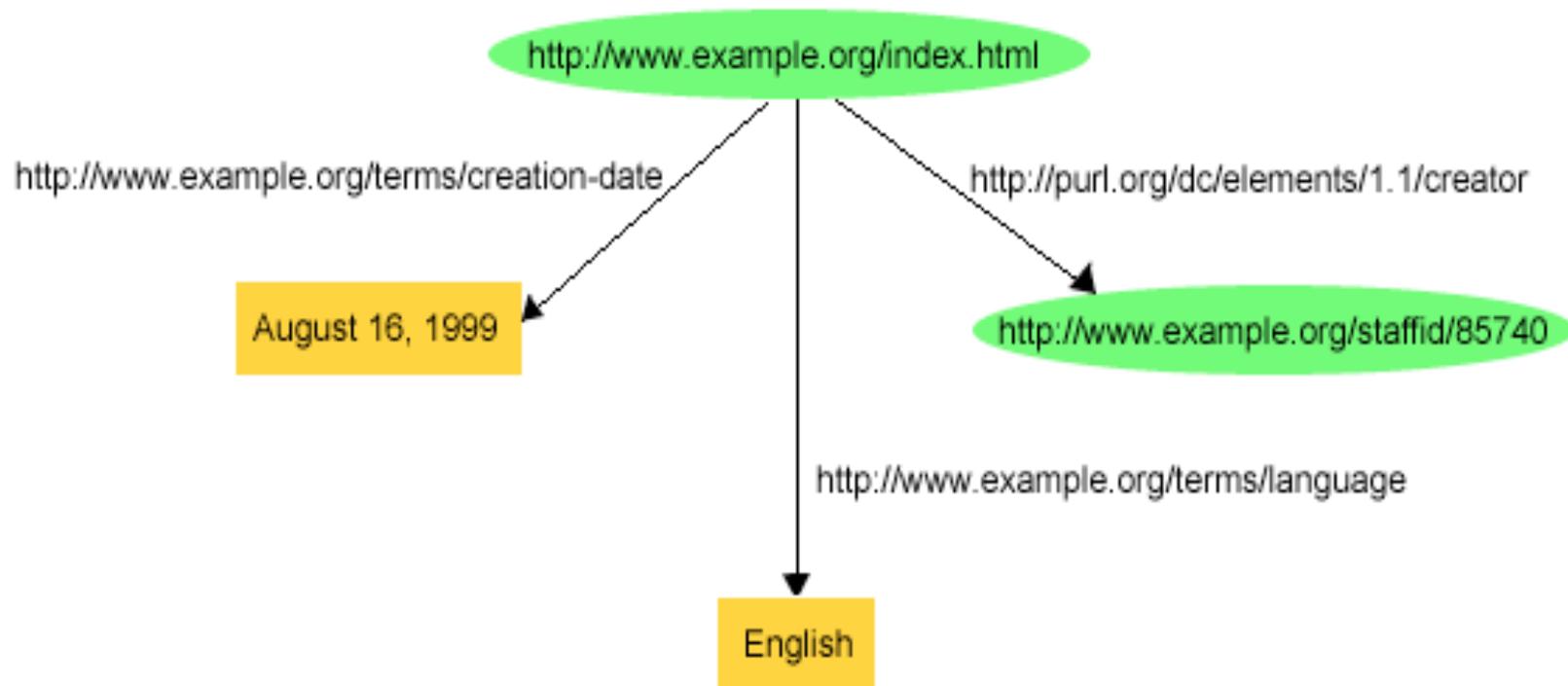
# Triple Example 1

- Every `Description` element describes a resource
- Every attribute or nested element inside a `Description` is a `property` of that Resource
- We can refer to resources by using URIs
- The object of a triple can also be a “literal” (a string)



# Triple Example 2

- **<http://www.example.org/index.html>** has a **creation-date** whose value is **August 16, 1999**
- **<http://www.example.org/index.html>** has a **language** whose value is **English**



# Triple Example 2

```
<?xml version="1.0"?>
```

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:exterms="http://www.example.org/terms/">
```

```
<rdf:Description rdf:about="http://www.example.org/index.html">
```

```
<exterms:creation-date>August 16, 1999
```

```
</exterms:creation-date>
```

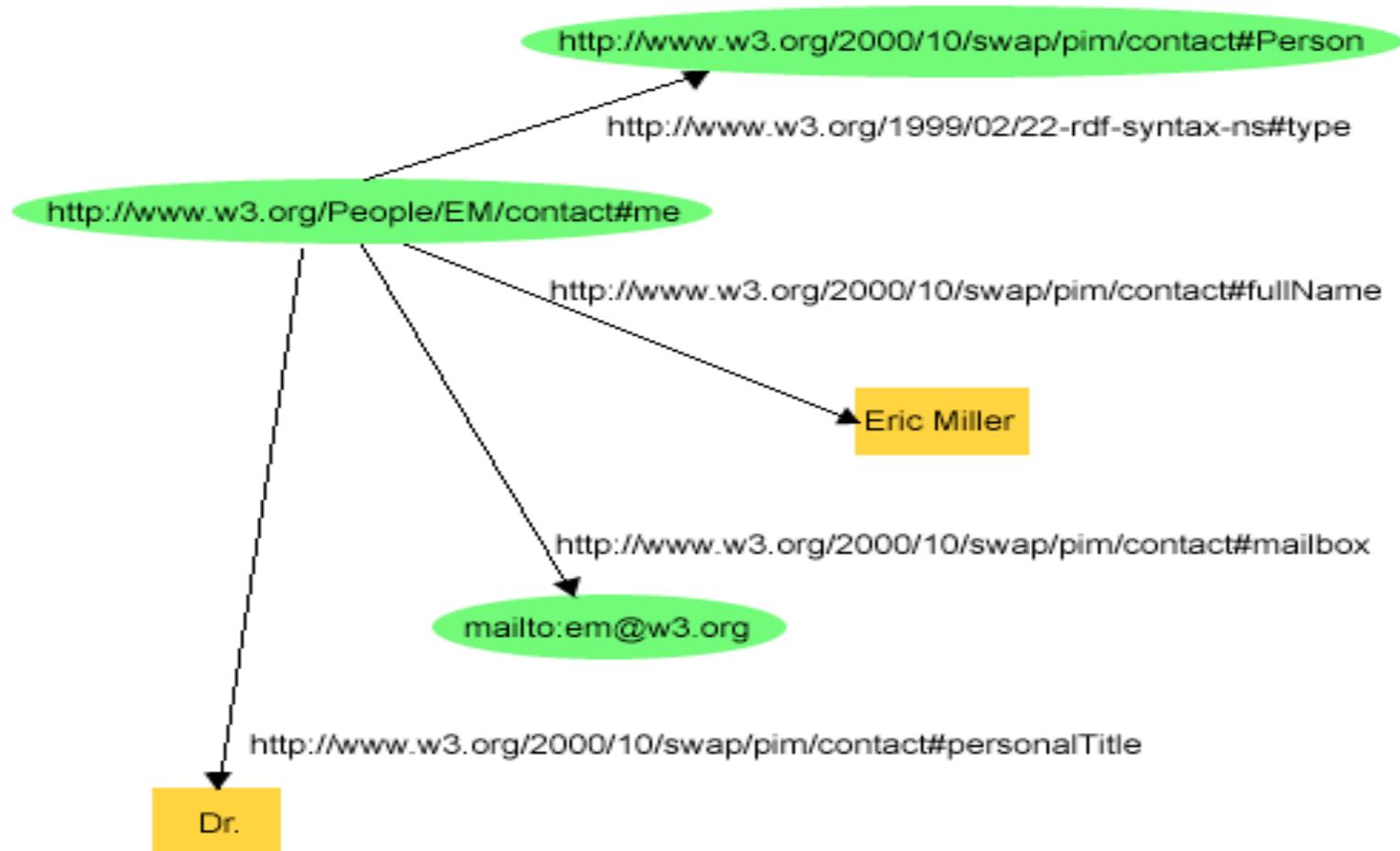
```
<exterms:language>English</exterms:language>
```

```
</rdf:Description>
```

```
</rdf:RDF> ♪
```

# Triple Example 3

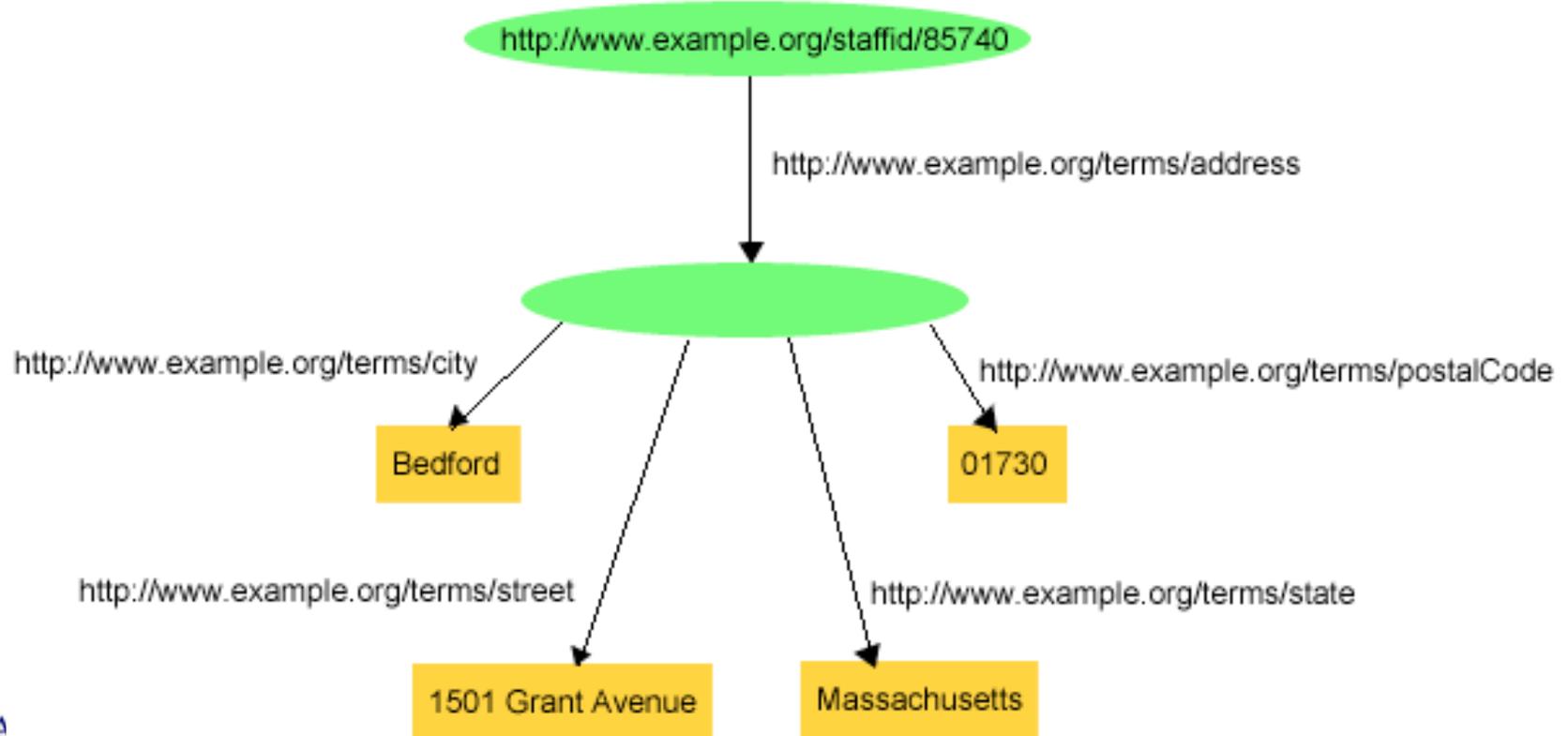
“There is a Person identified by <http://www.w3.org/People/EM/contact#me>, whose name is Eric Miller, whose email address is [em@w3.org](mailto:em@w3.org), and whose title is Dr.” ♪



# Triple Example 4

Address

ID	Street	State	City	ZIP
85740	1501 Grant Avenue	MA	Bedford	01730



# Some Well-known RDF Vocabularies

- FOAF (Friend-Of-A-Friend): describes **People**
  - Classes: Person
  - Properties: name, interest, mbox, schoolHomepage, workplaceHomepage
  - Namespace: <http://xmlns.com/foaf/0.1/>
- DOAP (Description-Of-A-Project): describes open-source **Project**
- DC (Dublin Core): describes **document**
  - Properties: title, creator, publisher, subject, ...
  - Namespace: <http://purl.org/dc/elements/1.1/>

# Eg, <http://pike.psu.edu/dongwon/foaf.rdf>

```
<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:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:admin="http://webns.net/mvcb/">
<foaf:PersonalProfileDocument rdf:about="">
  <foaf:maker rdf:resource="#me"/>
  <foaf:primaryTopic rdf:resource="#me"/>
</foaf:PersonalProfileDocument>
<foaf:Person rdf:ID="me">
<foaf:name>Dongwon Lee</foaf:name>
<foaf:title>Professor</foaf:title>
<foaf:givenname>Dongwon</foaf:givenname>
<foaf:family_name>Lee</foaf:family_name>
<foaf:mbox_sha1sum>12be8a0e055c93c16c3b01960a53ce4000d577d4</
  foaf:mbox_sha1sum>
<foaf:homepage rdf:resource="http://pike.psu.edu/dongwon"/>
<foaf:depiction rdf:resource="http://nike.psu.edu/dongwon/images/paris-face.jpg"/>
<foaf:phone rdf:resource="tel:814.865.0687"/>
<foaf:workplaceHomepage rdf:resource="http://www.psu.edu"/>
<foaf:workInfoHomepage rdf:resource="http://pike.psu.edu/dongwon/pro"/>
<foaf:schoolHomepage rdf:resource="http://www.cs.ucla.edu"/></foaf:Person>
</rdf:RDF>
```

Description about myself  
using FOAF syntax for  
crawling software

# Dublin Core

- <http://dublincore.org/>
- Started in 1995 at Dublin, Ohio
- A minimal set of descriptive elements that facilitate the description and the automated indexing of **document-like** networked objects, in a manner similar to a library card catalog
- Initially intended for resource discovery (eg, search engine)
- But expanded to be used with RDF

# Dublin Core

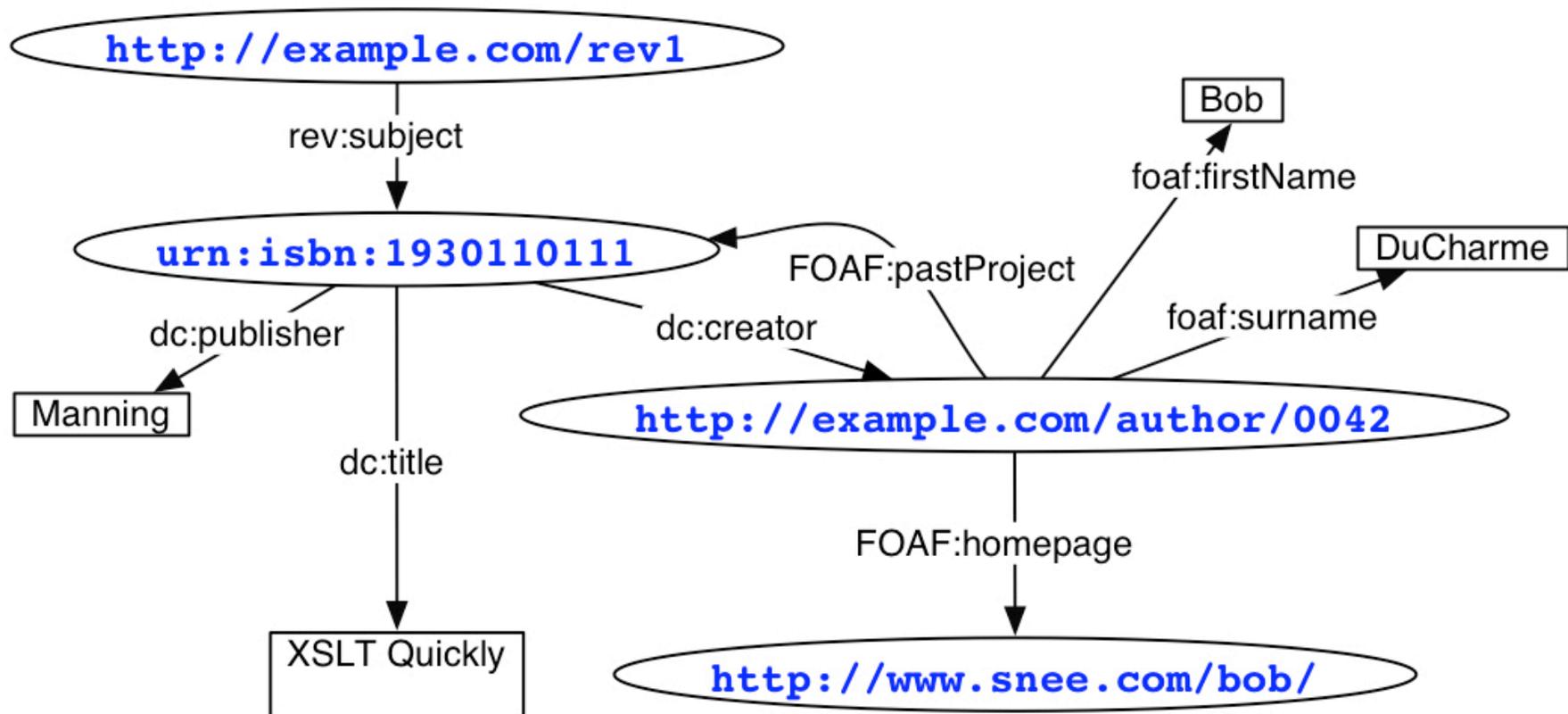
- **Properties (or elements) defined in Dublin Core**

- **Title:** A name given to the resource.
- **Creator:** An entity primarily responsible for making the content of the resource.
- **Subject:** The topic of the content of the resource.
- **Description:** An account of the content of the resource.
- **Publisher:** An entity responsible for making the resource available
- **Contributor:** An entity responsible for making contributions to the content of the resource.
- **Date:** A date associated with an event in the life cycle of the resource.
- **Type:** The nature or genre of the content of the resource.
- **Format:** The physical or digital manifestation of the resource.
- **Identifier:** An unambiguous reference to the resource within a given context.
- **Source:** A reference to a resource from which the present resource is derived.
- **Language:** A language of the intellectual content of the resource.
- **Relation:** A reference to a related resource.
- **Coverage:** The extent or scope of the content of the resource.
- **Rights:** Information about rights held in and over the resource.

# Eg, Using Dublin Core for IST516 Pages

```
<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="https://online.ist.psu.edu/ist516/">
    <dc:title>IST516 Web Page</dc:title>
    <dc:description>This web page is intended to be used for students
    registered for IST516, 2011.</dc:description>
    <dc:creator rdf:resource="http://pike.psu.edu/dongwon/foaf.rdf"/>
    <dc:date>2011-10-07</dc:date>
    <dc:subject>
      <rdf:Bag>
        <rdf:li>Course Introduction</rdf:li> ...
        <rdf:li>Course Schedule</rdf:li>
      </rdf:Bag>
    </dc:subject>
    <dc:format>text/html</dc:format> <dc:language>en</dc:language>
  </rdf:Description>
</rdf:RDF>
```

# Eg, Using RDF: DC + FOAF



## #2 RDF Schema (RDF-S)

- RDF gives a formalism for meta data annotation, and a way to write it down in XML, but it does **NOT** give any special **meaning** to vocabulary such as **subClassOf** or **type**
- RDF Schema allows you to **define vocabulary** terms
  - It gives “extra meaning” to particular RDF predicates and resources
  - This “extra meaning”, or semantics, specifies how a term should be interpreted

# RDF Schema Example

- RDF Schema terms (just a few examples):
  - Class, Property
  - type, subclassOf, subPropertyOf
  - range, domain
- These terms are the RDF Schema building blocks (constructors) used to create vocabularies:

<Person, type, Class>

<hasColleague, type, Property>

<Professor, subclassOf, Person>

<Lee, type, Professor>

<hasColleague, range, Person>

<hasColleague, domain, Person>

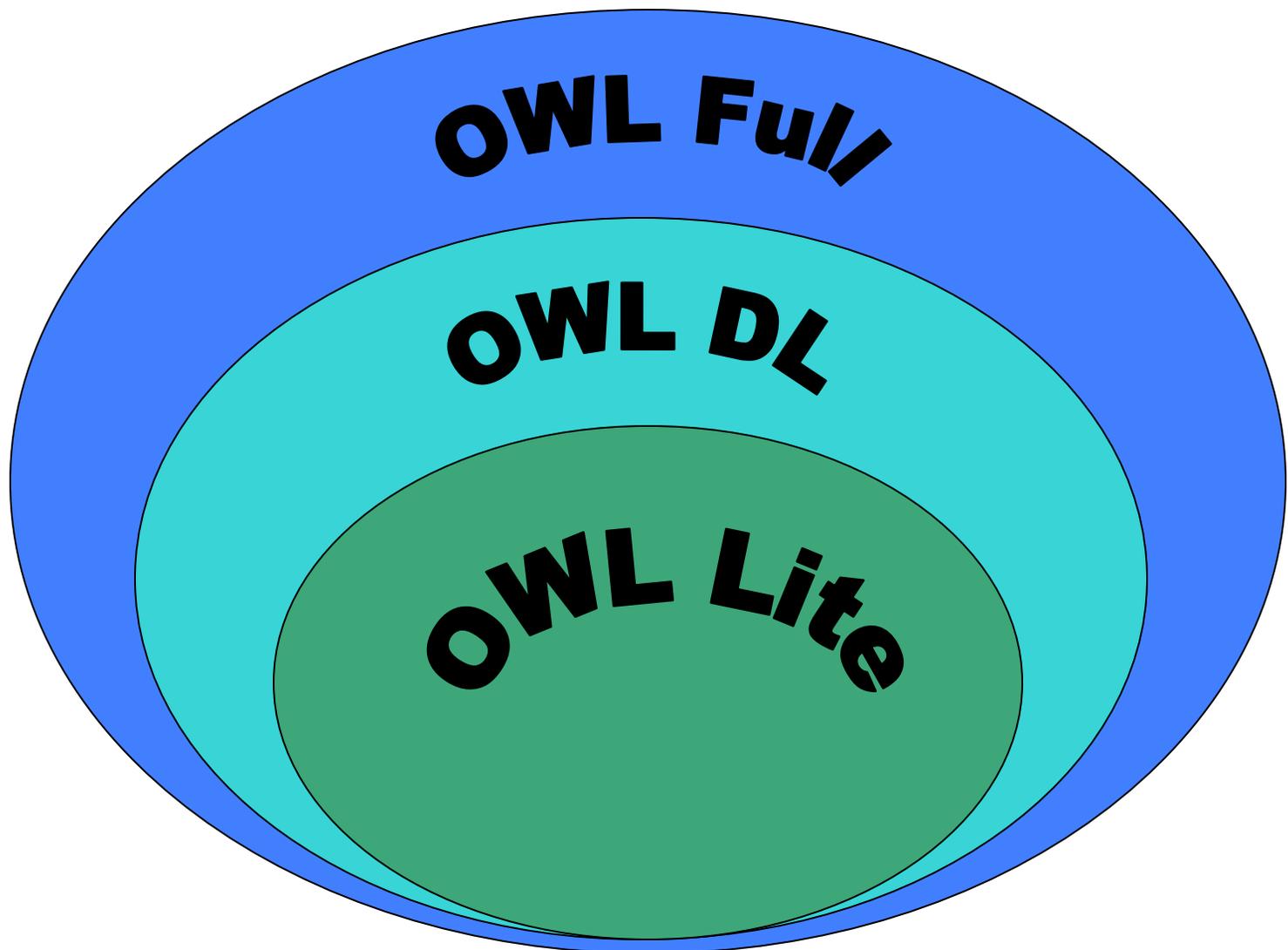
## #3 OWL (Web Ontology Language)

- OWL is built on top of RDF
- OWL is written in XML
- OWL is for processing information on the web
- OWL was designed to be interpreted by computers (ie, **not** for humans)
- OWL has three sublanguages
- OWL is a web standard

# Why OWL?

- XML tags values but doesn't provide "meaning"
- RDF / RDF Schema support a way of "modeling" data and their inter-relationships
- But, for machines to perform tasks on dynamic web content:
  - Go beyond RDF and formally and precisely describe the **meaning** of terminology on the Web
  - Describe logical relationships for machines to **reason** unspecified (but logical) relationships

# Three Sublanguages of OWL



# Three Sublanguages of OWL

- OWL Lite
  - Stripped down version for users needing to represent simple constraints
- OWL DL (Description Logics)
  - Allows further expression of desired constraints
  - Still certain restrictions: E.g., all results must be computable and able to be finished
- OWL Full
  - Most expressive version of OWL
  - There are no computational guarantees
  - W3C, *“It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full ...”*

# Basic OWL Lite Schema

- Individual (or Objects or Instances)
  - Eg, Dongwon Lee is an individual of Professor
- Class
  - Set of individuals (or objects)
  - Eg, The class Professor exists
- rdf:Property
  - States relationships between two individuals
  - Eg, hasStudent is a property of Dongwon Lee
  - Eg, hasAge is a property of Dongwon Lee, too

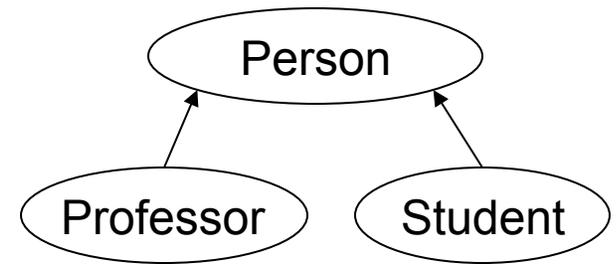
# OWL Heading

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<rdf:RDF
  xmlns:owl = "http://www.w3.org/2002/07/owl#"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-
ns#" xmlns:rdfs = "http://www.w3.org/2000/01/rdf-
schema#" xmlns:xsd = "http://www.w3.org/2001/
XMLSchema#"
  xmlns:us = "http://www.us.foo.com"
  xmlns:them = "http://www.foo.com">
<owl:Ontology rdf:about="">
<rdfs:comment>An example OWL ontology</
  rdfs:comment>
</owl:Ontology>
. . .
</rdf:RDF>
```



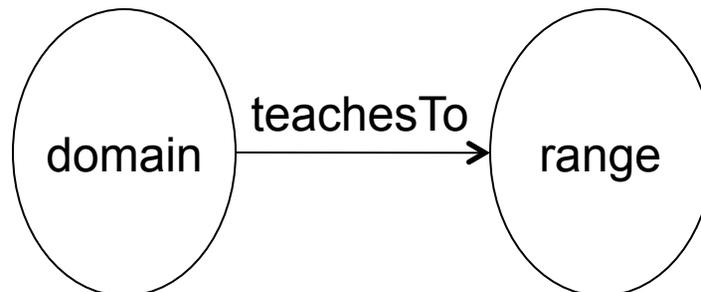
# Classes: <owl:Class>

- Standard Class Creation  
<owl:Class rdf:ID="Person"/>
- Sub-Class Creation  
<owl:Class rdf:ID="Professor">  
    <rdfs:subClassOf rdf:resource="#Person" />  
</owl:Class>  
<owl:Class rdf:ID="Student">  
    <rdfs:subClassOf rdf:resource="#Person" />  
</owl:Class>
- Individual of a Class Creation  
<Professor rdf:ID="Dongwon Lee"/>  
<Student rdf:ID="John Doe"/>



# Properties: <owl:ObjectProperty>

- Simple Property definition  
<owl:ObjectProperty rdf:ID="teachesTo"/>
- Same definition with some domain and range restrictions  
<owl:ObjectProperty rdf:ID="teachesTo">  
  <rdfs:domain rdf:resource="#Professor"/>  
  <rdfs:range rdf:resource="#Student"/>  
</owl:ObjectProperty>



# Transitivity Property

```
<owl:ObjectProperty rdf:ID= 'worksFor' >  
  <rdf:type rdf:resource="&owl;TransitiveProperty" />  
  <rdf:range rdf:resource= '#Person' />  
</owl:ObjectProperty>  
<Person rdf:ID= 'Graham Spanier' />  
<Person rdf:ID= 'Dongwon Lee' >  
  <worksFor rdf:resource= '#Graham Spanier' />  
</Person>  
<Person rdf:ID= 'John Doe' >  
  <worksFor rdf:resource= '#Dongwon Lee' />  
</Person>
```

- Due to the transitivity, the following does **NOT** need to be said:

```
<Person rdf:ID= 'John Doe' >  
  <worksFor resource= '#Graham Spanier' />  
</Person>
```

# Symmetric Property

```
<owl:ObjectProperty rdf:ID= 'worksWith'>  
  <rdf:type rdf:resource="&owl;SymmetricProperty" />  
  <rdfs:domain rdf:resource="#Person" />  
  <rdfs:range rdf:resource="#Person" />  
</owl:ObjectProperty>
```

```
<Person rdf:ID= 'New Student' >  
  <worksWith rdf:ID= '#John Doe' >  
</Person>
```

- Therefore, **NO** need to say

```
<Person rdf:ID= 'John Doe' >  
  <worksWith rdf:ID= '#New Student' >  
</Person>
```

# Using Protégé

- Protégé: a popular ontology editor to view, build, and reason ontologies
- Download latest version
  - <http://protege.stanford.edu/>
- One can use Protégé to:
  - View existing ontologies
  - Create new ontologies
  - Reason with ontologies



# Pets.owl (1/3)

Download from: <http://pike.psu.edu/classes/ist516/latest/labs/owl/pets.owl>

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://owl.cs.manchester.ac.uk/2009/07/sssw/pets#"
  xml:base="http://owl.cs.manchester.ac.uk/2009/07/sssw/pets"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:pets="http://owl.cs.manchester.ac.uk/2009/07/sssw/pets#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about=""/>
```



# Pets.owl (2/3)

```
<owl:Class rdf:about="#Animal"/>
```

```
<owl:Class rdf:about="#Cat">
```

```
  <rdfs:subClassOf rdf:resource="#Animal"/>
```

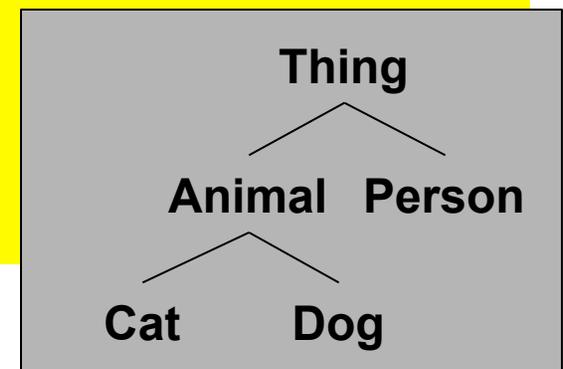
```
</owl:Class>
```

```
<owl:Class rdf:about="#Dog">
```

```
  <rdfs:subClassOf rdf:resource="#Animal"/>
```

```
</owl:Class>
```

```
<owl:Class rdf:about="#Person"/>
```



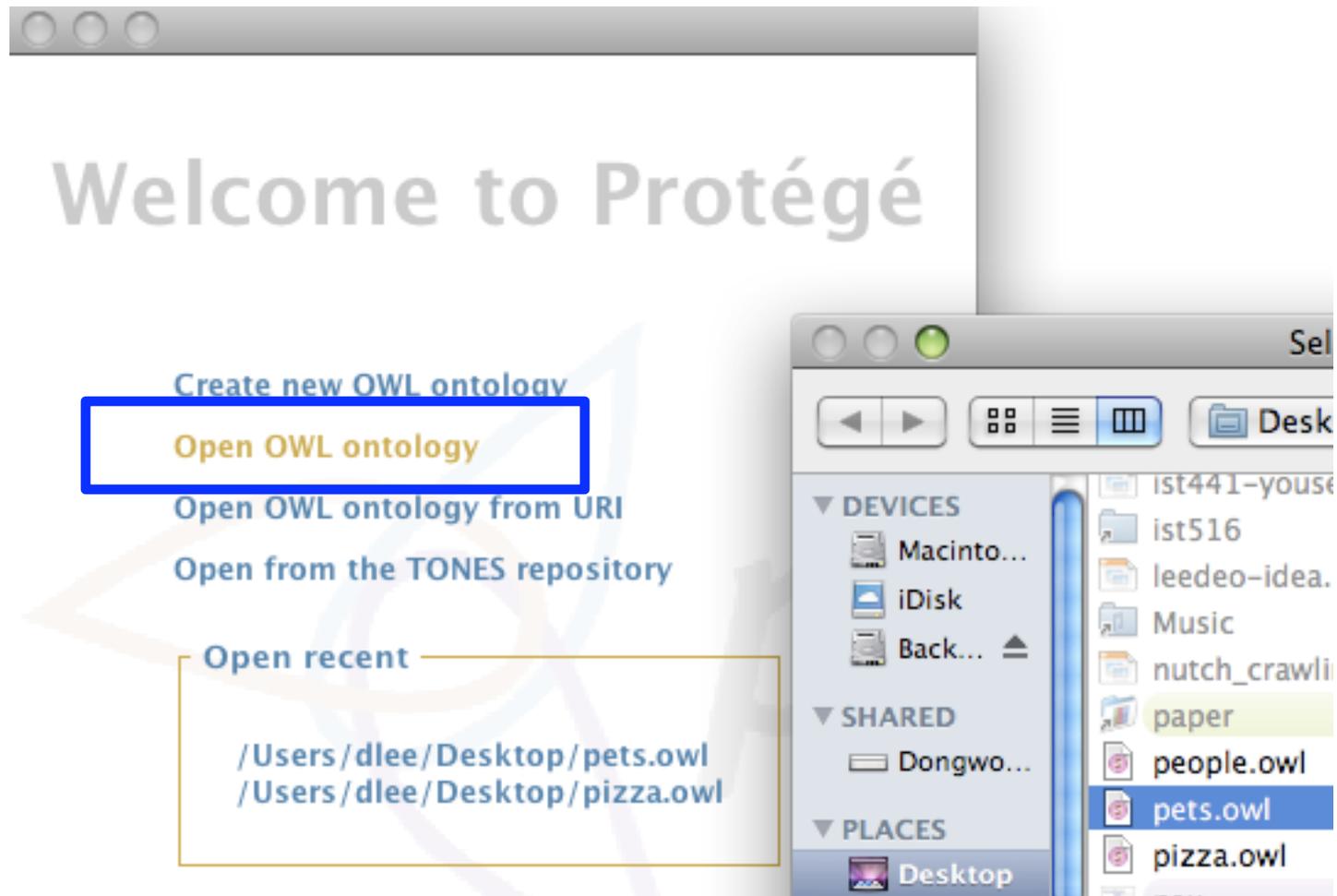
# Pets.owl (3/3)

```
<owl:ObjectProperty rdf:about="#hasPet">  
  <rdfs:range rdf:resource="#Animal"/>  
  <owl:inverseOf rdf:resource="#isPetOf"/>  
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:about="#isPetOf"/>
```

```
</rdf:RDF>
```

# 1. View Existing Ontologies



# 1. View Existing Ontologies

The screenshot displays the Protege ontology editor interface for the file 'pets (http://owl.cs.manchester.ac.uk/2009/07/sssw/pets)'. The 'Entities' tab is selected and highlighted with a blue box. The interface is divided into several panels:

- Class hierarchy (Dog):** A tree view showing the hierarchy: Thing (parent), Animal (child), Cat (child of Animal), Dog (child of Animal), and Person (child of Animal). The 'Dog' class is highlighted with a blue box, and the word 'CLASS' is written next to it.
- Object property hierarchy (hasPet):** A tree view showing the hierarchy: topObjectProperty (parent), hasPet (child), and isPetOf (child of hasPet). The 'hasPet' property is highlighted with a blue box, and the word 'PROPERTY' is written next to it.
- Usage: hasPet:** A panel showing 'Found 4 uses of hasP' with a list: hasPet InverseOf isPetOf, hasPet Range Animal, and Object Property hasPet.
- Characteristics:** A list of checkboxes for property characteristics: Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, and Irreflexive.
- Description: hasPet:** A panel showing 'Domains (intersection)' with 'Animal' listed, and 'Equivalent object properties' with 'isPetOf' listed.

At the bottom right, the status bar indicates 'No reasoner set' and 'Show Inferences' is checked. The Penn State logo is visible in the bottom left corner.

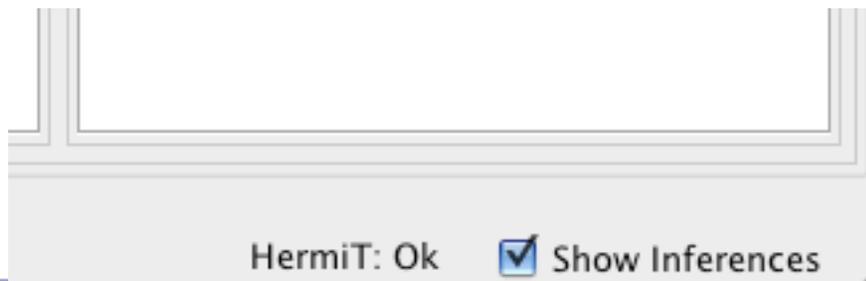
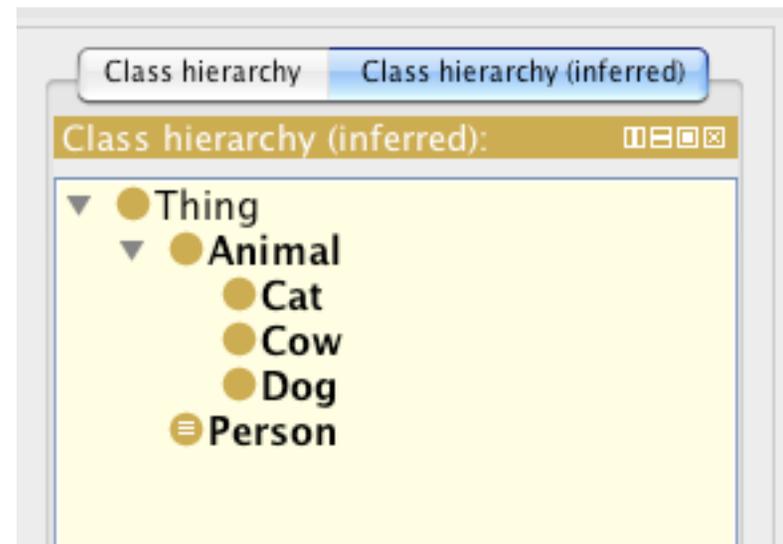
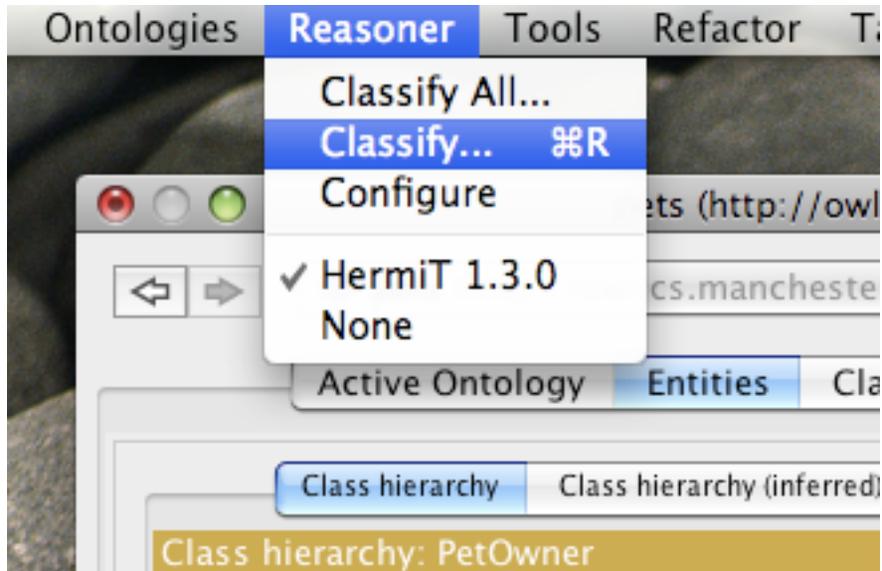
## 2. Create New Ontologies

- New Classes
  - Cow is a sub class of Animal
    - Cow, Cat, Dog are disjoint each other
  - DogOwner is a sub class of Person
    - Who has some Dogs
- New Individuals
  - “Fluffy” is an individual of Cat
  - “Happy” is an individual of Dog
  - “Lee” is an individual of Person who owns a Dog

# 2. Create New Ontologies

The screenshot displays the Protege OWL editor interface for an ontology named 'pets'. The browser address bar shows the URL: `http://owl.cs.manchester.ac.uk/2009/07/sssw/pets`. The main menu includes 'Active Ontology', 'Entities', 'Classes', 'Object Properties', 'Data Properties', 'Individuals', 'OWLviz', 'DL Query', and 'OntoGraf'. The 'Entities' tab is active, showing a class hierarchy for 'PetOwner' and an object property hierarchy for 'hasPet'. The 'Class hierarchy: PetOwner' panel shows a tree structure with 'Thing' as the root, followed by 'Animal', 'Person', 'Cow', 'Cat', 'Dog', and 'PetOwner'. The 'Object property hierarchy: hasPet' panel shows a tree structure with 'topObjectProperty' as the root, followed by 'hasPet' and 'isPetOf'. The 'Individual Annotations' panel shows the 'Lee' individual with two uses: 'Individual: Lee' and 'Lee Type PetOwner'. The 'Description: Lee' panel shows the type 'PetOwner'. The 'Property assertions: Lee' panel shows options for object property assertions, data property assertions, negative object property assertions, and negative data property assertions. The bottom status bar indicates 'HermiT: Ok' and 'Show Inferences' is checked.

# 3. Reason with Ontologies



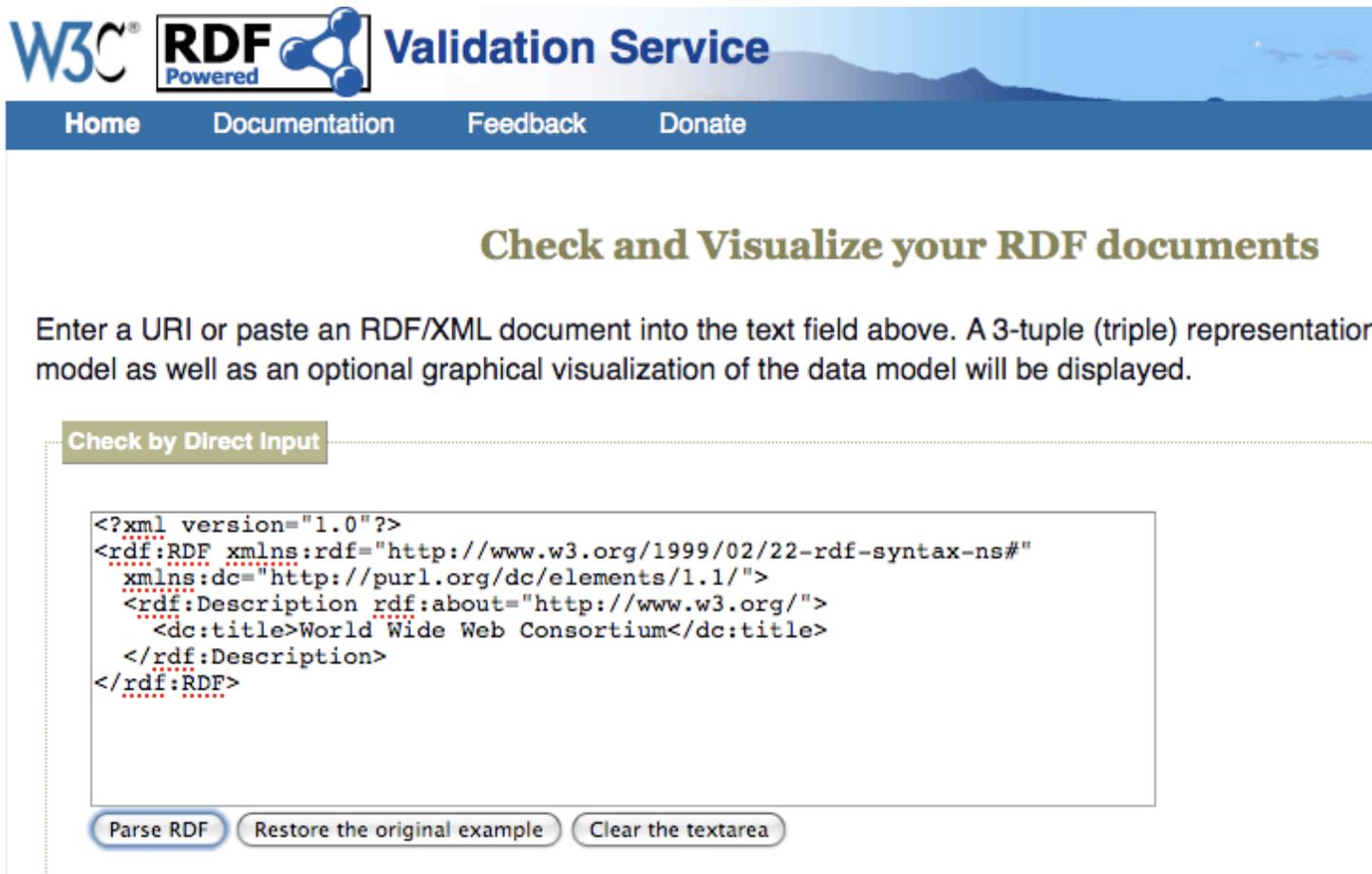
# More OWL Examples



- Wine
  - <http://www-ksl.stanford.edu/projects/wiris.txt>
- Meal
  - <http://www-ksl.stanford.edu/projects/wine/mc-s.txt>
- Wine Agent 1.0
  - <http://onto.stanford.edu:8080/wino/index.jsp>

# RDF/OWL Validator

- <http://www.w3.org/RDF/Validator/>
- OWL is type of RDF document



The screenshot shows the W3C RDF Validation Service website. At the top, there is a navigation bar with links for Home, Documentation, Feedback, and Donate. Below the navigation bar, the main heading reads "Check and Visualize your RDF documents". A paragraph of text instructs the user to enter a URI or paste an RDF/XML document into a text field above, which will then display a 3-tuple representation and an optional graphical visualization. A tab labeled "Check by Direct Input" is selected. Below this tab is a text area containing the following RDF/XML code:

```
<?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>
```

At the bottom of the text area, there are three buttons: "Parse RDF", "Restore the original example", and "Clear the textarea".

# References

- RDF Primer
  - <http://www.w3.org/TR/2003/WD-rdf-primer-20030905/>
- Introduction to the Semantic Web and RDF, A.M. Kuchling
  - <http://www.amk.ca/talks/2004-12-02/>
- Tutorial on the W3C OWL
  - <http://www.cs.bell-labs.com/cm/cs/who/pfps/talks/owl-tutorial/>
- OWL Web Ontology Language Overview
  - <http://www.w3.org/TR/owl-features/>
- Protégé Wiki
  - [http://protegewiki.stanford.edu/wiki/Main\\_Page](http://protegewiki.stanford.edu/wiki/Main_Page)
- Protégé OWL Tutorial
  - <http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/>

