Semantic Web and Python
Concepts to Application development

Vinay Modi
Voice Pitara Technologies Private Limited
Outline

• Web
• Need better web for the future
• Knowledge Representation (KR) to Web – Challenges
• Data integration – challenges
• KR to Web - solutions for challenges
• Metadata and Semantic Web – protocol stack
• RDF, RDFS and SPARQL basic concepts
• Using RDFLib adding triples
• RDFLib serialization
• RDFLib RDFS ontology
• Blank node
• SPARQL querying
• Graph merging
• Some possible things one can do with RDFLib
Text in Natural Languages

Multimedia

Images

Web

Deduce the facts; create mental relationships
Need better Web for the future

I Know What You Mean
KR to Web – Challenges

- Traditional KR techniques and Network effect
- Algorithmic complexity and Performance for information space like W3
- Scaling KR
KR to Web – Challenges

- Representational Inconsistencies
- Machine down Partial Information

Continue ... 1
Data integration - Challenges

• Web pages, Corporate databases, Institutions
• Different content and structure
• Manage for
  – Company mergers
  – Inter department data sharing (like eGovernment)
  – Research activities/output across labs/nations
• Accessible from the web but not public.
Data Integration – Challenges

• Example: Social sites
  – add your contacts every time.

• Requires standard so that applications can work autonomously and collaboratively.
What is needed

• Some data should be available for machines for further processing
• Data should be possibly combined, merged on Web scale
• Some time data may describe other data – i.e. metadata.
• Some times data needs to be exchanged. E.g. between Travel preferences and Ticket booking.
Metadata

• Data *about* data

• Two ways of associating with a resource
  – Physical embedding
  – Separate resource

• Resource identifier

• Globally unique identifier

• Advantages of explicit metadata

• Dublin core, FOAF
KR to Web – Solution for Challenges

Solve syntactic interoperability.
Standards

“Extra-logical” infrastructure.
Network effect

Scalable Representation languages

Semantic Web

Use Web Infrastructure

Continue ... 2
Semantic Web

Web extension

Information
- Machine readable
- Meta-data

Exchange
Integrate
Process
Machine automated

User interface and applications
- Trust

Proof
- Unifying logic

Querying: SPARQL
- Ontologies: OWL
- Rules: RIF/SWRL

Taxonomies: RDFS

Data interchange: RDF
- Syntax: XML

Identifiers: URI
- Character set: UNICODE

Cryptography
RDF basic concepts

- W3C decided to build infrastructure for allowing people to make their own vocabularies for talking about different objects.

- RDF data model:
RDF basic concepts

• RDF graphs and triples:

Subject

Predicate

Object

http://in.pycon.org/smedia/slides/semanticweb_Python.pdf
title
Semantic Web and Python

• RDF Syntax (N3 format):

@prefix dc: <http://purl.org/dc/elements/1.1/> .
RDF basic concepts

- Subject (URI)
- Predicate (Namespace URI)
- Object (URI or Literal)
- Blank Node (Anonymous node; unique to boundary of the domain)

Diagram:

- Subject: http://.../isbn/67239786
- Predicate: a:publisher
- Object: Addison-Wesley
- Predicate: a:name
- Object: Boston
- Predicate: a:city
RDF basic concepts

- Ground assertions only.
- No semantic constraints
  - Can make anomalous statements
RDFS basic concepts

• Extending RDF to make constraints
• Allows to represent extra-knowledge:
  – define the terms we can use
  – define the restrictions
  – What other relationships exist
• Ontologies
RDFS basic concepts

- Classes
- Instances
- Sub Classes
- Properties
- Sub properties
- Domain
- Range
SPARQL basic concepts

• Data
  @prefix foaf: <http://xmlns.com/foaf/0.1/> . 
  _:a foaf:name "Vinay" .
  _:b foaf:name "Hari" .

• Query
  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
  SELECT ?name
  WHERE { ?x foaf:name ?name . }

Results (as Python List)
["Vinay", "Hari"]
SPARQL basic concepts

• Query matches the graph:
  – find a set of variable -> value bindings, such that result of replacing variables by values is a triple in the graph.
• SELECT (find values for the given variable and constraint)
• CONSTRUCT (build a new graph by inserting new values in a triple pattern)
• ASK (Asks whether a query has a solution in a graph)
RDFLib

- Contains Parsers and Serializes for various RDF syntax formats
- In memory and persistent graph backend
- RDFLib graphs emulate Python container types – best thought of a 3-item triples.
  \[
  \{\text{subject, object, predicate}, \text{(subject, object, predicate)}, \ldots\}
  \]
- Ordinary set operations; e.g. add a triple, methods to search triples and return in arbitrary order
RDFLib – Adding triple to a graph

```python
from rdflib.Graph import Graph
from rdflib import URIRef, Namespace

inPyconSlides = Namespace("http://in.pycon.org/smedia/slides/"")
dc = Namespace("http://purl.org/dc/elements/1.1/")

g = Graph()
g.add((inPyconSlides['Semanticweb_Python.pdf'], dc:title,
       Literal('Semantic Web and Python – concepts to application development'))
```
RDFLib – adding triple by reading file/string

```python
str = """@prefix dc: <""" + dc + """> .
    @prefix inPyconSlides : <""" + inPyconSlides + """> .
inPyconSlides :'Semanticweb_Python' dc:title 'Semantic Web and Python – concepts to application development' . ""

from rdflib import StringInputSource
rdfstr = StringInputSource(str)
g.parse(rdfstr, format='n3')
```
RDFLib – adding triple from a remote document

```
inPyconSlides_rdf = 'http://in.pycon.org/rdf_files/slides.rdf'
g.parse(inPyconSlides_rdf, format='n3')
```
Creating RDFS ontology


<http://in.pycon.org/hasSlidesAt> rdf:type rdfs:Property.

<http://in.pycon.org> rdfs:label 'Python Conference, India'
RDFLib – SPARQL query

• Querying graph instance
  
  # using previous rdf triples

  \[q = "'"\text{PREFIX dc: <http://purl.org/rss/1.0/>}
  \text{PREFIX inPyconSlides : <http://in.pycon.org/smedia/slides/>}
  \text{SELECT ?x ?y}
  \text{WHERE \{ ?x dc:title ?y. \}}
  '''\]

  \text{result = g.query(q).serialize(format='n3')}
from rdflib import BNode

profilebnode = BNode()

http://.../delegate/vinaymodi

hasProfile

hasName

hasTutorial

hasCompany

http://in.pycon.org/.../.../Sematicweb_Python

http://www.voicepitara.com

Vinay Modi
RDFLib – graph merging

g.parse(inPyconSlides_rdf, format='n3')
g1 = Graph()
myns = Namespace('http://example.com/')

# object of the triple in g1 is subject of a triple in g.
g1.add((http://vinaymodi.googlepages.com/',
      myns['hasTutorial'], inPyconSlides['Semanticweb_Python.pdf']))
mgraph = g + g1
RDFLib – some possible things you can do

- Creating named graphs
- Quoted graphs
- Fetching remote graphs and querying over them
- RDF Literals are XML Schema datatype; Convert Python datatype to RDF Literal and vice versa.
- Persistent datastore in MySQL, Sqlite, Redland, Sleepycat, ZODB, SQLObject
- Graph serialization in RDF/XML, N3, NT, Turtle, TriX, RDFa
End of the Tutorial

Thank you for listening patiently.

Contact:

Vinay Modi
Voice Pitara Technologies (P) Ltd
vinay@voicepitara.com

(Queries for project development, consultancy, workshops, tutorials in Knowledge representation and Semantic Web are welcome)