**Python for RDF**

There are many methods to search for information on the internet. Using python with its libraries and the advantages of using semantics can be a good way to search and represent data. In this topic, I will use the python as a tool to build an application to search job information. Moreover, I will use protégé to build a taxonomy and setup a few rules that are necessary for look up for an appropriate job that one wishes.

Thus, the main components of this project are python, its libraries, protégé to build RDF/OWL file, and online database. Also, this project is based on the book *Programming the Semantic Web* by Toby Segaran, Colin Evans, and Jamie Taylor, O'Reilly, 2009. Since this book was written a quite long time ago, some of its references do not work or do not exist anymore. I will try to look for other resources.

**The path to this project**.

Semantic web is a great way to model data, especially it can work pretty well with a huge amount of the information with different format types from different resources. With triple format of the semantic (RDF/owl), semantic web can unify and link those data in the most efficient way. Moreover, with the assistance of python, a powerful programming language, data can be searched, filled and modeled very flexibly and clearly.

Python and its libraries that are used to build RDF.

First of all, what are some RDF API for Python? These are a few API for python that support and build RDF:

### RdfAlchemy (<http://www.openvest.com/trac/wiki/RDFAlchemy>)

* Fuxi (<http://code.google.com/p/fuxi/>)

### ORDF ([ordf.org](http://ordf.org/))

### Django-rdf (<http://code.google.com/p/django-rdf/>)

### Djubby (<http://code.google.com/p/djubby/>)

### Redland (<http://librdf.org/>)

### SuRF (<http://packages.python.org/SuRF/>)

Those API above have different character to support and build RDF in python; however, for the project, the RDFLib library is the main source to use for its features such as: parsers and serializers for RDF/XML, N3, NTriples, N-Quads, Turtle, TriX, RDFa and Microdata store implementations for in memory storage and persistent storage on top of the Berkeley DB, a SPARQL 1.1 implementation - supporting SPARQL 1.1 Queries and Update statements. (<https://rdflib.readthedocs.org/en/latest/> or <http://rdflib.readthedocs.org/en/3.2.0/gettingstarted.html> )

The steps to build an application to search information:

1. Searching for the websites that have the database for this project.

Some of the references in this book are not relevant at this time. The example that the Obama administration working to build an application to search for job can be used. I follow this example to find more information from other websites that have the database for the search application.

1. Try out some codes as a model example for this project, testing and modifying according to the design of this project. To do this I have to use Wing IDE to test and adding codes that need for the project. Also I have to test with some tabular data examples, a list of the jobs and its titles for example to see it works or not.
2. Building up taxonomy with the rules that fit the examples of this project. I need to study more on the graph and the triplestore to make the file RDF/OWL to bring into python and program it. This step also sets rules for which particulate information or data that one wants for look for. For example, with an veteran who has a limited physical health, there should be jobs that suit their needs,…
3. Setup the initial dataset for Sesame and Cherry in python. There are many instruction on other websites that mention the use of these two python package. After studying them I will run codes with the RDF/OWL files.
4. Setting up a web application server: this step is also followed the step above. The Cherry package will help to build web application.
5. Creating HTML templates for viewing the result of the search engine application.
6. Building the queries across many datasets: this step will take many programming codes. It also depends on the specific example of what jobs one looks for.
7. This project will be an open develop one. Its functionalities can be expanded to other use for example to build dataset from public sources, its data can be used for other semantic applications and can be modeled according to a specific need. Building an exhibit in different format also can be done as an extension of this project.

With python background, yet I can have a big problem with this project when RDF needs to be brought in to model data. So, taxonomy and bringing it to python can be time consuming for me. However, if this project can be done well, I will understand the use of semantic web better and then I can be able to expand skill of building RDF with other programming language such as Java and other API. The outcome expectation of this project is to build an application and its basic functionality that can run for a short time. It can be flexible to new data and can share it at a certain level.