

RDF transportable organization databases

Leonardo Maycotte

March 31, 2004

Abstract

The purpose of this writing is to present a methodology of how to describe and transport your organization information across the internet with the use of Protégé[1].

1 First step

Defining your organization schema

We use the protégé classes as if they were sql tables and the protégé slots as if they were the column and row names. Next is presented an example of a Protégé class with its slots and slot values.

Protégé class: `employee_personal_info`

Protégé slots and slot values:

Name; Charles

Last_Name; Babbage

ID; 101

Phone; 8183255886

Email, `cbabbage@mail.com`

Like this we define n protégé classes of our organization schema with one slot in common, the ID slot. This is our key slot used to define relational information in our other protégé classes, where each ID represents the unique personal number of an employee.

2 Second step

Defining organization information

Once we have defined our organization schema in protégé, we save it and make it public as a rdfls file on the url provided as our project namespace.

Now the organization schema is available for use by anyone to import it to their local protégé installation and use it to define instances based on the organization schema. The instances defined by everyone are expected to be saved and sent as rdf files to an application that makes specific employee data queries.

3 Third Step

Parsing the rdf files

The application for parsing and making queries on the rdf files is a web application programmed in python using 4Suite, which is an open-source platform for XML and RDF processing[2]. The rdf docs are parsed as objects with bounded methods where the rdf statements (subject, predicate and object triplets) are stored in the document object and you can see them with this simple python function:

```
from Ft.Rdf import Util
from Ft.Rdf.Util import *

#the rdf and rdfs files are made with prot\`eg\`e
file = 'organization.rdf'
schema = 'organization'
rdfs = 'organization.rdfs'

#paths depend on server file locations
path = '/var/www/html/umbrella/' +file
path2 = '/var/www/html/umbrella/' +rdfs

#namespaces map used for queries
NSS = {'rdf': "http://www.w3.org/1999/02/22-rdf-syntax-ns#" ,
       'dc': "http://purl.org/dc/elements/1.1/",
       'sy': "http://purl.org/rss/1.0/modules/syndication/",
       'rss': "http://purl.org/rss/1.0/",
       'vsort': "http://rdfinference.org/versa/0/2/sort/",
       'a': "http://protege.stanford.edu/system#",
       'rdfs': "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#",
       'empresa': "http://10.17.134.162/umbrella/empresa#",
       'agenda': "http://10.17.134.162/umbrella/agenda#"
      }
```

```

#m is the python object that contains our rdf file
m, driver = DeserializeFromUri(path)
#d is our python object that contains our rdfs file
d, driver = DeserializeFromUri(path2)

#to print all the rdf statements of our rdf and rdfs files
print m.statements()
print d.statements()

#to print in order the subject, predicate and object of our rdf file
for stmt in document.statements():
    print stmt.subject, stmt.predicate, stmt object

```

Now we are ready to make queries on our rdf files.

4 Fourth Step

Making queries

The queries are made with VERSA, an RDF query language[3]. First we make available the information that can be found thanks to the properties defined on our rdfs file and then the user searches by name the information he wants. For example, if the properties of our rdfs file are email and phone (there are many more), the user can search for George email or phone and he will get all the email addresses and telephone numbers of the people named George on the organization rdf file available. The application is a web application (python cgi's) that searches by name and last names the information defined on the rdfs file. The queries are made with VERSA as relational searches using the ID slot value and the protégé instance values.

References

- [1] "The protégé ontology editor and knowledge acquisition system." Internet site. <http://protege.stanford.edu/>.
- [2] "4suite: an open-source platform for xml and rdf processing." Internet site. <http://4suite.org/index.xhtmll>.
- [3] U. Ogbuji, "Versa, the rdf query language." Internet site. <http://uche.ogbuji.net/tech/rdf/versa/>.