ESW07 Assignment I

- View
- Edit
- Track

Overview

The curricula for Bachelor- und Masterstudien der Informatik at the Technische Universitaet Wien have been published as LaTeX documents.

All information therein about particular subjects, their titles, modalities and other specifics, into which topical focus they fall, etc. has been managed with flat text files, mainly for authoring reasons.

Your first task is

- to analyze this data from an RDF/S perspective;
- write (A) a converter to generate an RDF stream for the course data;
- (B) put it into an RDF store;
- and (C) enrich the store with an appropriate RDFS schema.

Once this is accomplished, you will provide (D) a simple web service which can be used to extract some course information in XHTML form.

The Data

All data you will find in the `dat` directory of the package This `.tgz` file also contains there a description file (`spezifikation.txt`, in German) which details what the individual fields mean.

All the LaTeX boilerplate you can safely ignore.

The Platform

All code has to be produced in Perl (or alternatively in Ruby), so that you can rather freely choose your favourite platform. The only other constraint is that you have to use the Redland RDF Libraries.

If you prefer **not** to develop on your own platform, I will provide you with a VMware appliance which you (hopefully) will be able to use in the lab or any machine which runs `vmplayer`. That appliance is using Debian Etch (with some parts from unstable) with all the necessary software installed. The only downside of this approach is that the appliance is rather large (several hundred MBs).

A: Converting into RDF

The program which takes all the individual text files and munges them into an RDF stream has to be called `sal2rdf` and should be invoked like this:

perl sal2rdf informatik.yaml

`informatik.yaml` (attached) is the configuration file detailing which files should actually be consumed. Its format is YAML and it is also attached here already for your convenience.

The `sal2rdf` program (also attached) should use the following boiler plate:

```
use constant {
   RDF => "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
# other namespaces might go here
   };

use Config::YAML;
my $cfg_file = shift @ARGV
        or die "config file expected as sole parameter";
my $config  = Config::YAML->new (config => $cfg_file);

use XML::Writer;
my $writer = new XML::Writer(NAMESPACES => 1,
                     PREFIX_MAP => { main::RDF => 'rdf' },
                     FORCED_NS_DECLS => [ RDF ]);
$writer->startTag([RDF,'RDF']);
$writer->characters("Here my instance data goes!");
$writer->endTag([RDF,'RDF']);
$writer->end();
```

Obviously, all output goes to `STDOUT`. To test whether the output is well-formed XML you can invoke it within a pipe:

perl sal2rdf informatik.yaml | xmllint -

Once this works, you may test your instance data against the RDF Validating service.

Some notes:

- At **NO** point your program is allowed to create any temporary files. This is **EXTREMELY** bad style.
- As the data involved is quite complex, I do **NOT** expect you to migrate all aspects of it into RDF. You can limit yourself to those parts which are necessary to satisfy (D). Still, you are not allowed to loose data, so keep any unstructured data in a text (comment) property.
- The original data (like **all** data) is unclean, so you may have to fix some minor representation problems. Note that XML does not like to have special characters like <, >, & or quotes and apostrophes in (parsed) data segments.

B: RDF Store

The script to load an RDF stream into the RDF store should be invoked like this:

..... | perl rdf2store

So it has to consume from RDF content from `STDIN`. Which store technology you choose is up to you, but hashes in `"hash-type='bdb'"` should work

just fine for our purposes.

C: RDF Schema

To give your instance data a bit more meaning, create a schema authored in Turtle format, so that you can embed it **directly** into the `rdf2store` program.

In Perl you can embed it within the `DATA` pseudo file:

```
undef $/;
my $rdfs = <DATA>;

print $rdfs;
__DATA__
aaa bbb ccc .
aaa bbb ddd .
aaa bbb eee .
```

It mayb be ugly, but it is convenient. Like everything in Perl.

D: Web Service

For the web service you will need a simple HTTP daemon. It is to be started simply via

perl informatik

and needs only to support the following requests URIs:

- For a particular course, for instance `/lva/EfSemW`: The result should be an XHTML document containing all relevant information about that course.
- List courses for a particular Pruefungsfach, for instance `/pruefungsfach/Kke/`: The result should be a simple XHTML list representing the list of course codes for that Pruefungsfach.

Add simple error handling and return status `404` for not found objects.

To extract this information from the RDF store, you should use SPARQL, at least to the degree it is supported by Redland Rasqal. It should be sufficient for our purposes here.

Supporting Documentation

We will cover most of it in the lectures at some point:

- RDF Primer (RDF/XML)
- Redland programming in Perl (soon to be updated, Redland librdf, API)
- SPARQL Introduction (SPARQL)

Parting Notes

Everything not specified here, you can (have) to decide on your own. Please document your (design) decision appropriately in the code.

Attachment	Size
TUW-Inf-2007.tgz	179.44 KB
sal2rdf	975 bytes
informatik.yaml	311 bytes

- 6 reads

Posted In

- esw07

Submitted by rho on Sun, 10/07/2007 - 19:42.
Home

tags in rho's universe

apache Australia Austria Bond University conference data living perl tmql topic maps turing test web sites
more tags

Recent blog posts

- Tutorial: RDF Redland Perl Programming
- TMQL: Last Changes and Issues
- Net::Citadel 0.01 Released
- Q & A (Part VII)
- Content Landscape (Part I)
- Q & A (IV, V and VI)
- You're Ugly as Hell
- Q & A (Part III)
- The Nifsan Affair
- WIAWTD: Topic Maps Ontology Language (TMOL)

more

Upcoming events

- TMRA 2007(1 day)
- Vienna.pm TechSocialMeet:(28 days)


more

Weekly archives

- 10/07/07 - 10/13/07
- 09/30/07 - 10/06/07
- 09/23/07 - 09/29/07
- 09/16/07 - 09/22/07
- 09/09/07 - 09/15/07
- 09/02/07 - 09/08/07
- 08/26/07 - 09/01/07
- 08/19/07 - 08/25/07
- 08/12/07 - 08/18/07

Popularity

All time:

- Tarpitting with Apache and mod_security2
- reverse-proxying ntop: Tragedy in 4 Acts
- Do despair.com !
- Semantic Wikis on Vicodin (Steroids)
- Sacrifices to the Deans

Administration

- Administer:
  - Content management...
  - Site building...
  - Site configuration...
  - User management...
  - Logs...
  - Help

rho

- Archives
- My blog
- Current Polls
- Search
- Recent posts
- My account
- Create content...
- Log out

Advertising

kill -9 `/dev/cat`

"Trust me", the white-haired man said...

- No Way Out

Let evolution finish its work.