Lab Assignment 2: Semantic Web with Jena and Dublin Core

Goal: *Be able to create in-memory and persisted models with Jena.*

1. Install "Gruff: A Grapher-Based Triple-Store Browser for AllegroGraph" from Franz:
   a. Get the latest **standalone** version (as of writing, Gruff for AllegroGraph 3.3, second half of page) for your operating system at:
      http://www.franz.com/agraph/gruff/index.lhtml#download
      i. You have to provide a name and an e-mail address.
      ii. You do not need to separately download Allegrograph.
      iii. You should review the Installation Instruction on the download page.
   b. You may find additional documentation in the "doc.html" file.
   c. Complete the Gruff Worksheet, save your worksheet as:
      Lab2_1_<YourID>.<suffix>
      in one of the allowable formats in the table below where:
      <YourID> should be replaced with your first initial and lastname.
      <suffix> should be replaced as follows:

Allowable Output Formats	Description
.doc	Microsoft Word
.rtf	Microsoft Rich Text Format

2. Create an in-memory Jena model (the default model)

   Presume your favorite director is Stanley Kubrick. Represent the following information in a new "FILM" vocabulary that you create (preferably, first on paper). The information you wish to represent in inter-related knowledge form is:
   - Movies – Dr. Strangelove, A Clockwork Orange
   - Year – go to imdb.com to find this.
   - Represent Kubrick himself by using the vCard vocabulary.

   a. The graph of data must be fully connected (i.e., no un-connected blobs/pieces).
   b. There must be some type of person and movie classes – and each movie or person must belong to an appropriate class (i.e., you make up URI's for these classes and then use *rdf:type* as the verb between the objects and these classes).
   c. There must be no blank nodes.
   d. Each URI that you create should begin with the namespace:
      http://utdallas/semclass#
   e. While not required for success, you may want to use methods available in the model class like those in the list below. You should reference the Jena documentation for details:

- `model.createProperty()`
- `model.createResource()`
- `model.createResource(URI_string).addProperty()`

This part of the lab (i.e., part 2) is not submitted for grading. It is, however, the foundation for the next part.

3. Persist the graph using Dublin Core in Jena and TDB

All Stanley Kubrick movies are based on books. “Dr. Strangelove” was based on the book “Red Alert” and “A Clockwork Orange” was based on the book with the same name.

Extend part 2 to include the books and the authors of the books. Use the Dublin Core vocabulary to represent the books and vCards to represent the authors. Add the relationship between movies, directors, books, and authors. Store the results in your default TDB triple store (using same storage locations as Lab #1).

a. The same criteria apply as for part 2.
b. Use Dublin Core to provide metadata about books (i.e., links from books are Dublin Core properties unless they are a property you create or to a class).
c. Use vCard to represent data about authors (i.e., links from authors are vCard properties unless they are a property you create or to a class).
d. Output the model in the following formats to the indicated filename in the default execution directory:

Output Format	Output filename*
1) RDF/XML	Lab2_3_<YourID>.xml
2) N3	Lab2_3_<YourID>.n3

* <YourID> should be replaced with your first initial and lastname.

Submission:

Please compress your final Eclipse project into one zip file and submit the zip file on eLearning. Name your main java file “Lab_2.java”. Include any files you created for compiling and running your program. Include the Gruff worksheet and model files in the zip file as well.

Grading (100 points):

-100	Nothing submitted
-25	Gruff worksheet not submitted
-10	Gruff worksheet in unallowed format
-5 each	Gruff worksheet individual answers wrong
-50	Jena program not submitted
-4	Object URI’s (in subject-predicate-object triples) are literals, not resources (i.e., a literal is not copied everwhere—it is declared once and referenced elsewhere as needed)
-10	Doesn’t Run
-3	Has Log4j runtime complaint
-10	Lacking vCard data in program and result file
-10	Lacking Dublin Core data in program and result file
-5	Incorrect output filenames
-5	Incorrect source filenames
-10	Lacking use of TDB