

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 version (as of writing, Gruff for AllegroGraph 7.4.0) 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 should review the Installation Instruction on the download page.
 - b. You also need to separately download Allegrograph, either
<https://franz.com/agraph/downloads/server> for a linux system or get a virtual machine at <https://franz.com/agraph/allegrograph/vm/>.
 - i. Use can use the online Allegrograph at <http://gruff.allegrograph.com:10035>, but this should only be used for testing.
 - c. You may find additional documentation in the “doc.html” file.
 - d. Complete the Gruff Worksheet, save your worksheet as:
Lab2p1_<YourID>.<suffix>
in one of the allowable formats in the table below where:
<YourID> should be replaced with your UTD NetID.
<suffix> should be replaced as follows:

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

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

Presume your favorite writer is H. G. Wells. Represent the following information in a new “BOOK” vocabulary that you create (preferably, first on paper). The information you wish to represent in inter-related knowledge form is:

- Books – The War of the Worlds, The Time Machine
 - Year – go to Wikipedia to find this (the books).
 - Represent Mr. Wells 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 book classes – and each book or person resource 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 resources 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, 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 part 3.

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

Several H. G. Wells books were made into TV movies. The book “The War of the Worlds” made into a radio broadcast and two movies with the same name. The book “The Time Machine” was made into two movie with the same name.

Extend part 2 to include the radio broadcast and movies including the writer for the radio broadcast and directors of the movies. Use the Dublin Core vocabulary to represent the radio broadcast and movies and vCards to represent the writer and directors. Add relationships between books, author, movies, and writer/directors. Store the results in your default TDB triple store (using same storage locations as Lab #1).

- The same criteria apply as for part 2. Classes for movies and directors are made the same way.
- Use Dublin Core to provide metadata about the braodcast and movies (i.e., links from movies are Dublin Core properties unless it is a property you create).
- Use vCard to represent data about the writer/directors (i.e., links from writer/directors are vCard properties unless it is a property you create).
- Output the model in the following formats to the indicated filename in the default execution directory:

Output Format	Output filename*
1) RDF/XML	Lab2p3_<YourID>.rdf
2) N3	Lab2p3_<YourID>.n3

* <YourID> should be replaced with your UTD NetID.

- e. Check you model with Gruff.

Submission:

Please compress your final Eclipse project into one zip file and submit the zip file on eLearning. Name your main java file “Lab2.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
- 10 Summary Report of Program
- 4 Object URI's (in subject-predicate-object triples) are literals, not resources (i.e., a literal is not copied everywhere—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