TASK2- Implementing A Simple Information Processing Task with UIMA SDK

UIMA-Eclipse

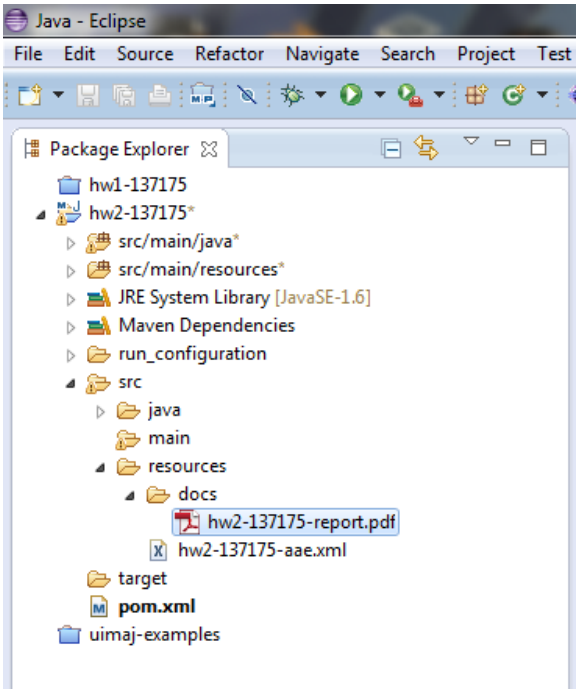
Student: Tania Patiño

ITAM

The structure of the project explained on the requirements:

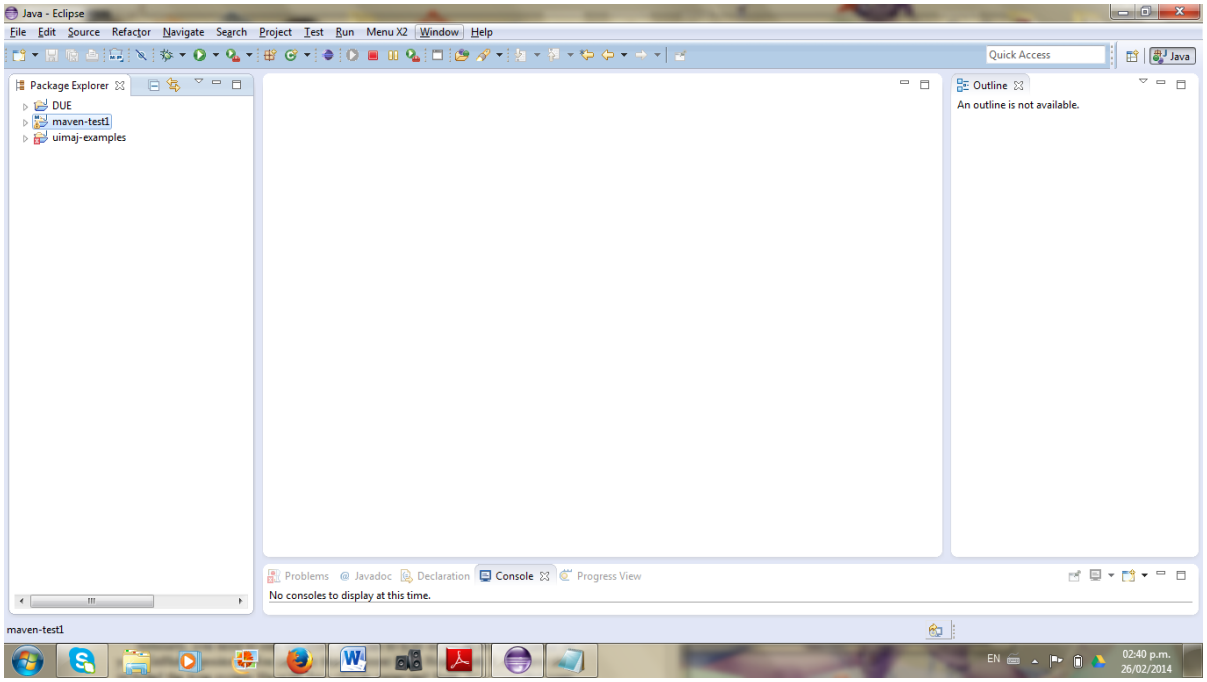
hw2-ID
- |- pom.xml
- ‘- src
- ‘- main
- |- java
- | ‘- **/*.java /* Java classes generated by JCasGen
- | and your UIMA annotators */
- ‘- resources
- |- hw2-ID-aae.xml /* your aggregate analysis engine */
- |- **/*.* /* analysis engine and other resources */
- ‘- docs
- ‘- hw2-ID-report.pdf /* your report for design */

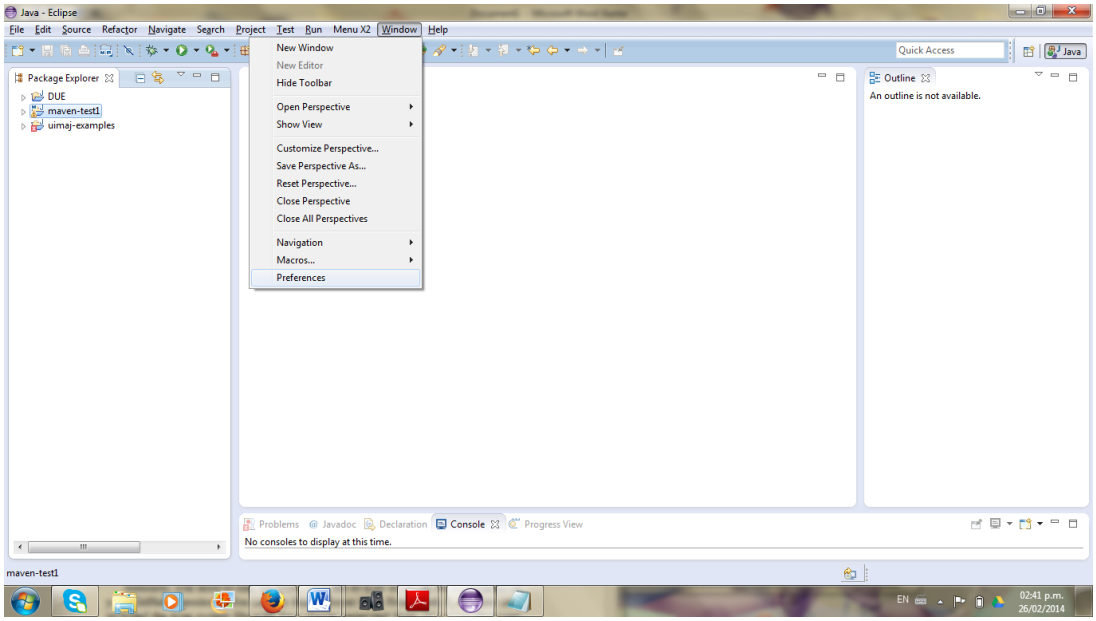
The structure of the project on Eclipse:



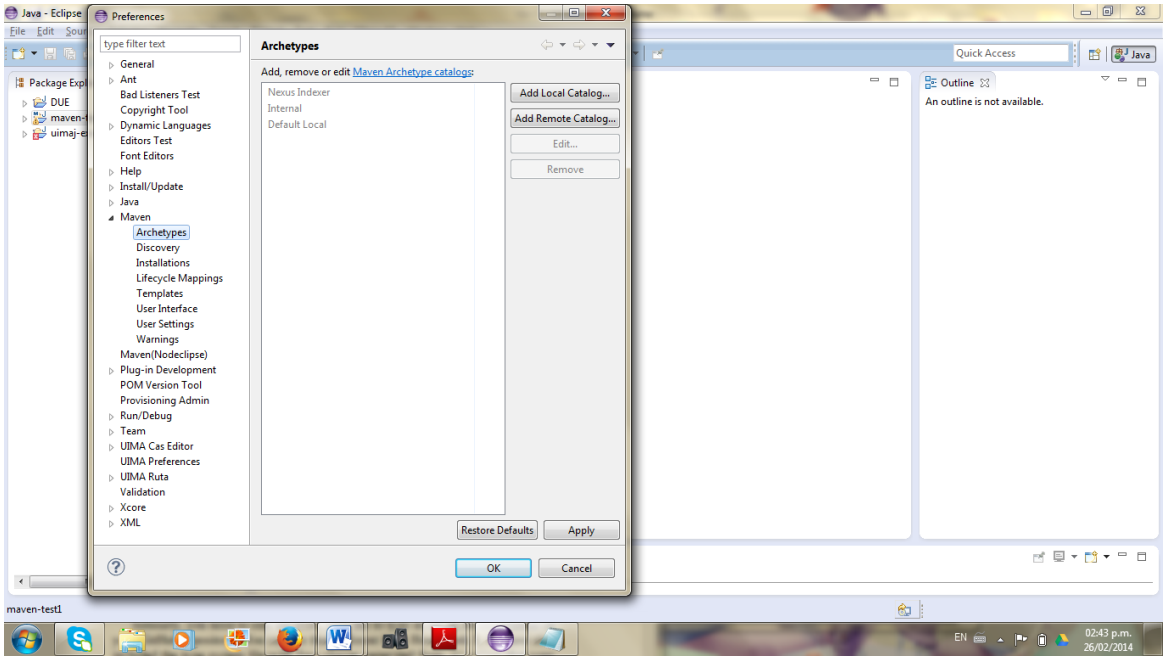
Task 1.1 Creating Maven project from the archetype

1. Creating a new archetype that will be added to the project go to the menu and select Window>Preferences.

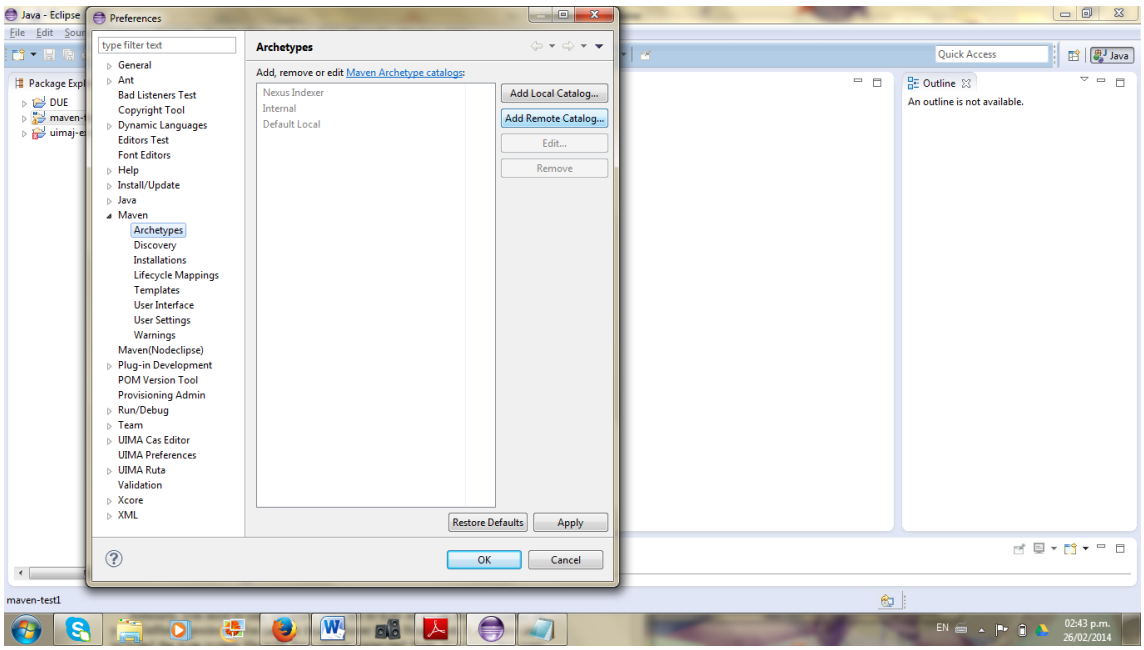




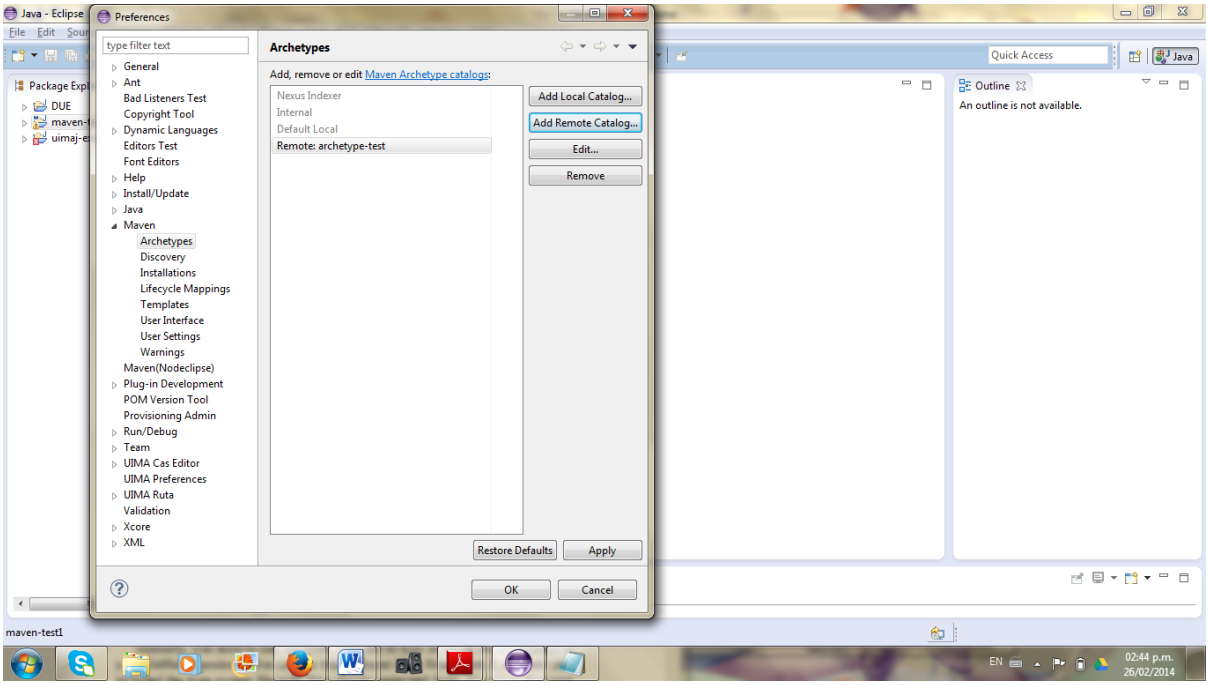
2. Selecting option Maven then Archetypes and then Add Remote Catalog as you can see in the next two images.



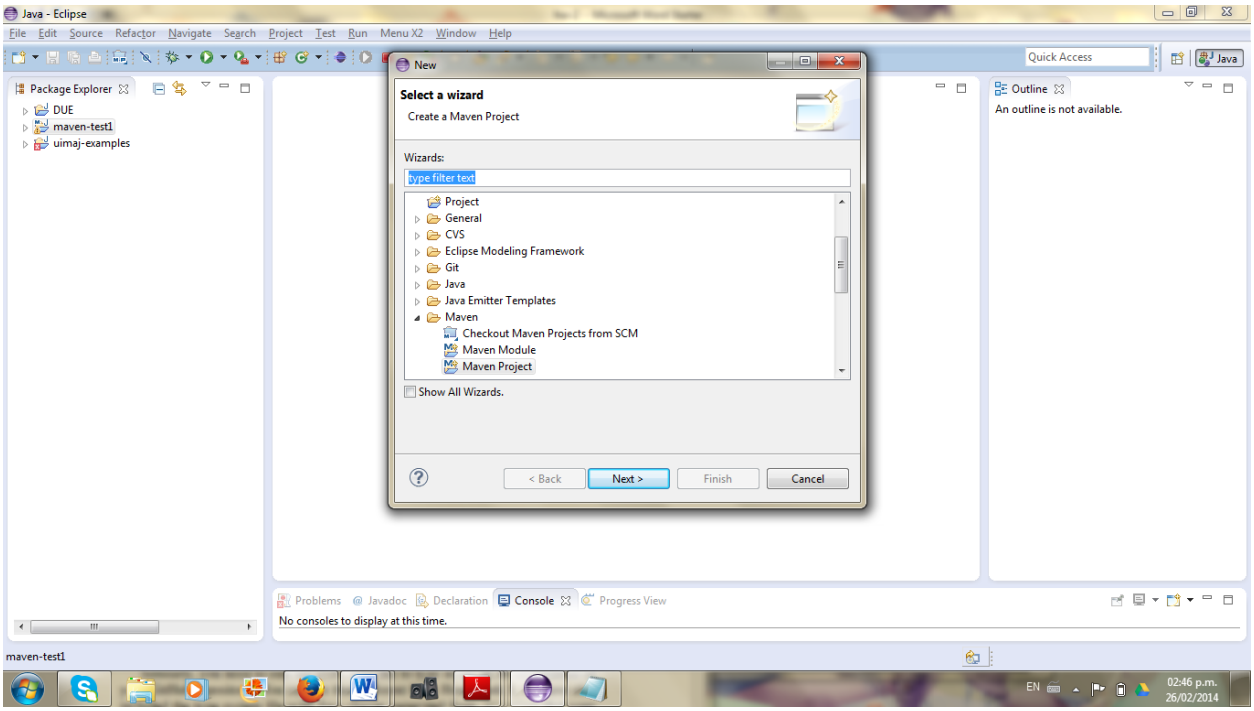
Selecting option Add remote catalog



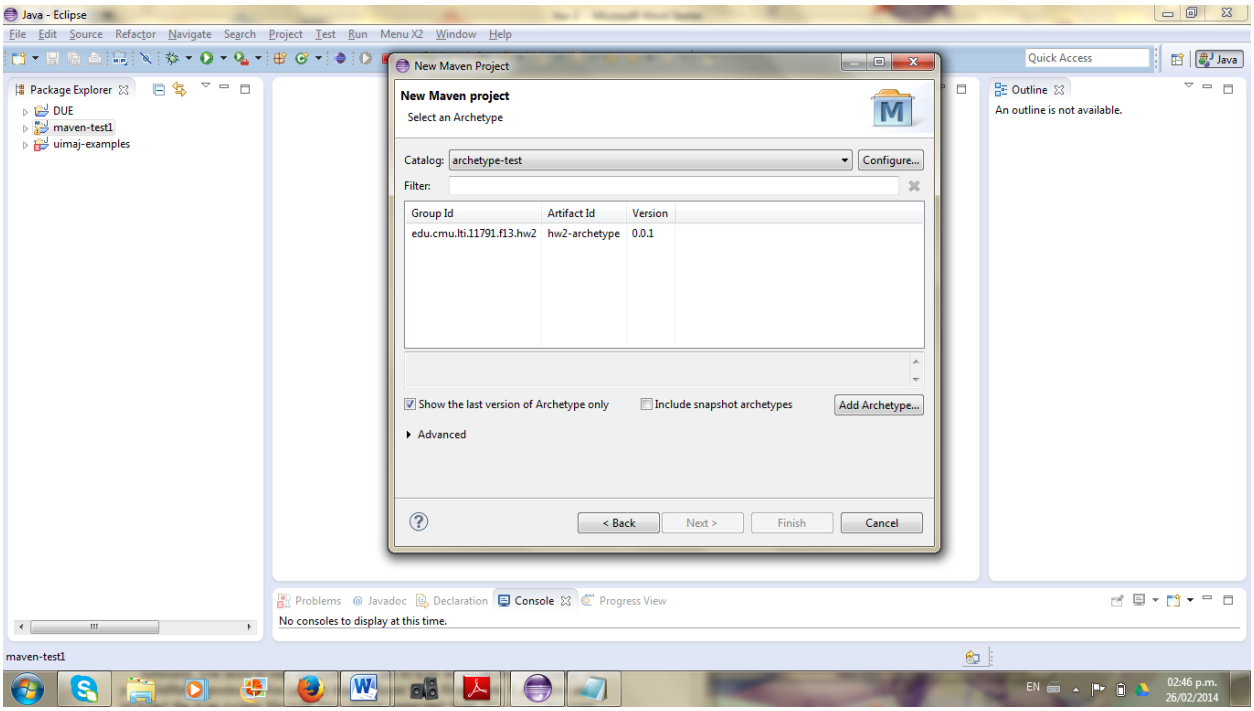
3. Creating archetype called archetype-test



4. When finish the last steps then with the right button of the mouse choose option New> Other> Maven Project and click in Next.

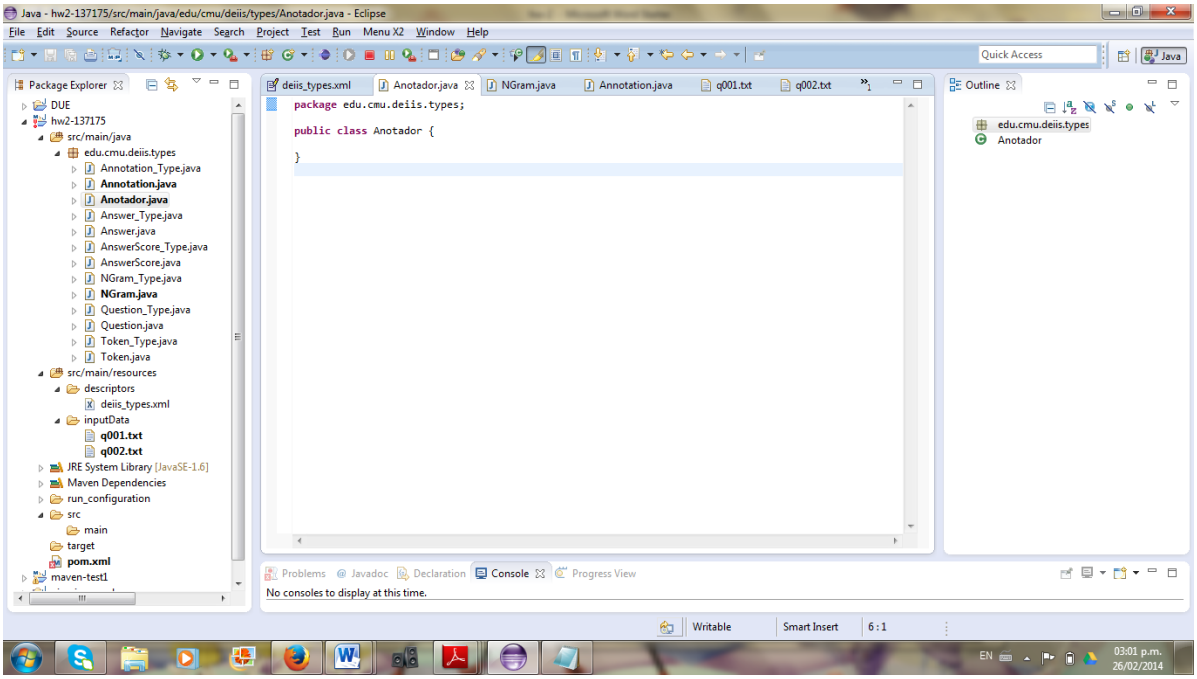


5. Select in the Catalog list the name of the annotator that was created and click on Finish, in this case the name is archetype-test where the GroupID is edu.cmu.lti.11791.f13.hw2 and ArtifactID is hw2-ID.



6. The final step for this task is to createsome java classes where objects should be called that communicate with the java classes that comes in the standard project like Objects of Annotation_Type.java, Answer.java, NGram.java, etc.

   First was created a class called Anotador.java but many more have to be created in this project.



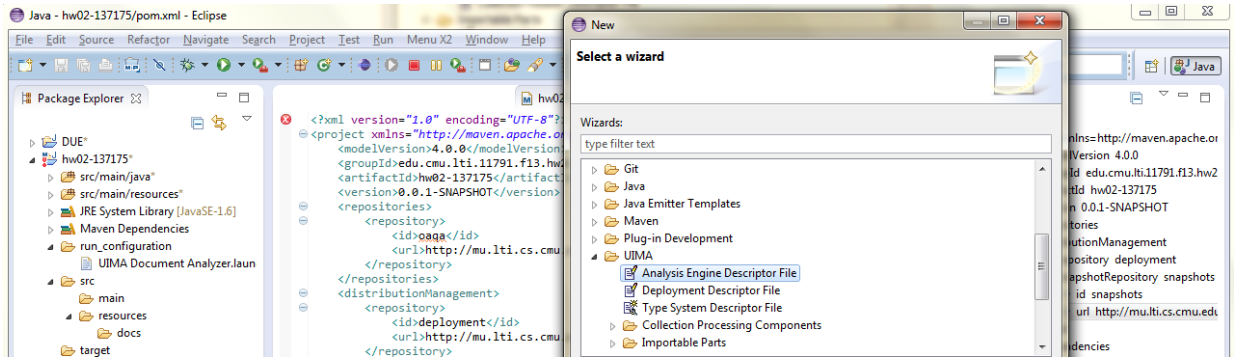
Main idea: create objects that call methods from other classes and those could analyze the files q001.txt and q002.txt.

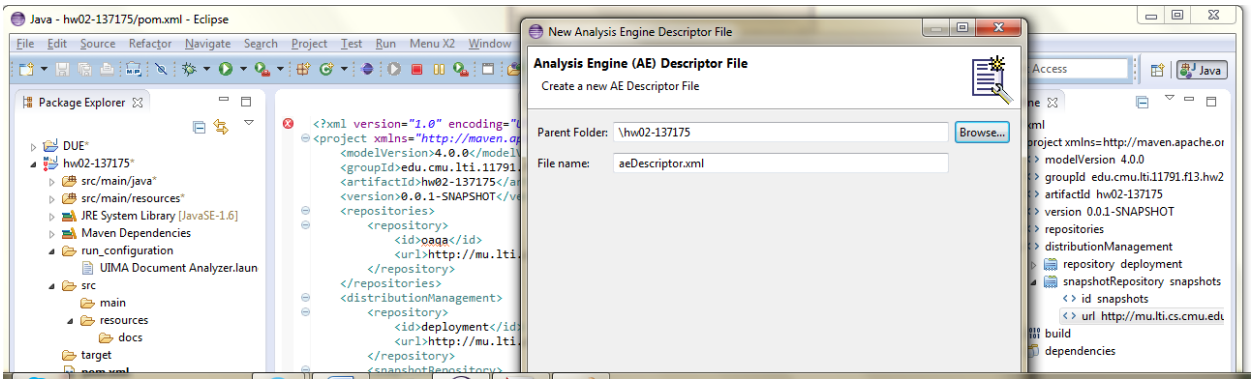
Task 1.2 Adding analysis engines for your pipeline

Description: to implement an aggregated analysis engine (hw2-ID-aae.xml) with several analysis engines that annotate the inputs and evaluate the performance of the aggregate analysis engine by comparing the system outputs with the gold-standard outputs.

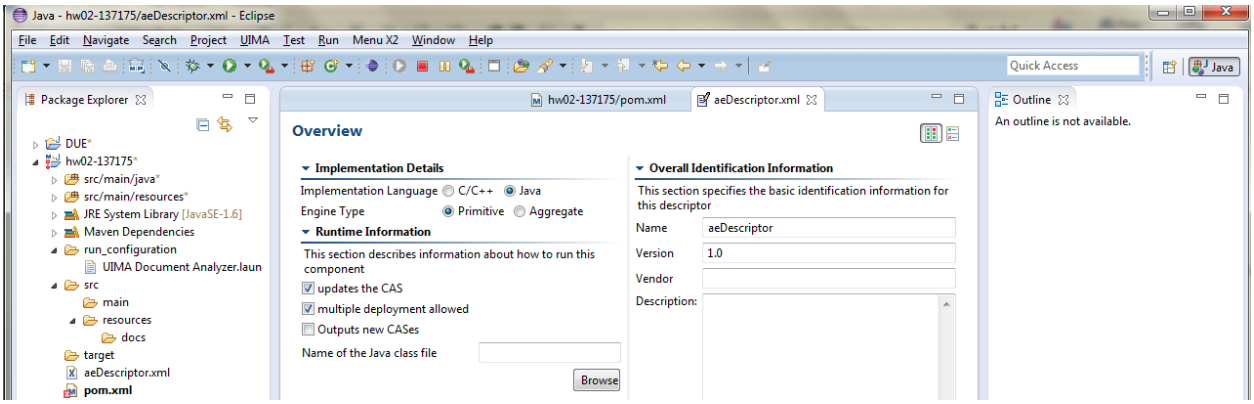
File: hw2-137175-aae.xml this is the analysis engine that contains a simple implementation.

1. Creating a New AE Descriptor
   File →New→ Other... menu





Description of the Component Descriptor Editor



Task 1.3 The Processing Pipeline

Main idea is to create five classes each one analyze:

1. Test Element Annotation: The system will read in the input file as a UIMACAS and annotate the question and answer spans. Each answer annotation will also record whether or not the answer is correct.

2. Token Annotation: The system will annotate each token span in each question and answer (break on whitespace and punctuation).

3. NGram Annotation: The system will annotate 1-, 2- and 3-grams of consecutive tokens.

4. Answer Scoring: The system will incorporate a component that will assign an answer score annotation to each answer. The answer score annotation will record the score assigned to the answer.

5. Evaluation: The system will sort the answers according to their scores, and calculate precision at N (where N is the total number of correct answers).

Note: in this task just a simple codification was made related to points1 and 2. The others points 3,4 and 5 are not coded in this task.