COM-35702 Design and Engineering of Intelligent Information Systems

Execution Architecture with CPE and Deployment Architecture with UIMA-AS

Important dates

- **Hand out: March 7.**[a]
- **Turn in: March 28.** In this homework, you will excute your pipeline from Homework 2 with a CPE and deploy it as a service with UIMA-AS. You should organize your project in the same hierarchy as shown below:

```
hw3-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 of homework 2*/
         |- hw2-ID-aae-client.xml /* clinet descriptor of your AAE */
         |- hw2-ID-aae-deploy.xml /* deployment descriptor of your AAE */
         |- hw3-ID-aae-as-CPE.xml /* CPE descriptor to test your AAE service */
         |- hw3-ID-CPE.xml        /* CPE descriptor of your homework 2 pipeline */
         |- scnlp-ID-client.xml   /* clinet descriptor for the remote UIMA-AS service *
         |- **/*.*                /* analysis engine and other resources */
         `- docs
            `- hw3-ID-report.pdf  /* your report for design */
```

[a]This version was built on March 21, 2014

Several notes about organizing your Maven project and other additional information:

1. **Submission:** The same way as you did for Homework 0, 1 and 2 (set up GitHub repo, create Maven project, write your code, submit to Maven repo), except that the name has changed to hw3-ID.

2. **Your report for design:** We expect to see how you design the execution and deployment architecture for the sample information system in your report. We will pull out your documents from your jar files. Remember to include your ID as part of file names, and put your name and ID in your document. Please submit in PDF format only.

3. **Javadocs:** Please remember to give an appropriate description for each annotator you create.

4. Please post your questions regarding Homework 3 on Piazza `https://piazza.com/itam.mx/spring2014/com35702`. For other issues or concenrs, you can also send us mails to, Elmer Garduno (`elmer.garduno@itam.mx`).

Task 1

Execution Architecture with CPE

In this task, you will need to learn what a Collection Processing Engine (CPE) is and how to use it. You are required to run your pipeline with a CPE instead of the UIMA Document Analyzer.

Task 1.1 Learning CPE

There are several useful resources to help you get started with CPE:

- You can learn basic concepts and usage about CPE from *Chapter 2. Collection Processing Engine Developer's Guide* (`http://uima.apache.org/d/uimaj-2.4.0/tutorials_and_users_guides.html#ugr.tug.cpe`).
- Also, you can view the manual for CPE GUI (`http://uima.apache.org/d/uimaj-2.4.0/tools.html#ugr.tools.cpe`).

Task 1.2 Creating and Running your CPE (25 pts)

1. (10 pts) Ideally, depending on where the input file is located (on a local file system, in a jar file, or from an external sources, e.g. network), your collection reader needs to be general enough to establish a connection to the file source and open a stream to read the content. But for our task, you only need to consider a folder of files located on the file system, which means you can use org.apache.uima.tools.components.FileSystemCollectionReader directly, and only need to write a Collection Reader descriptor to fit your needs.
2. (10 pts) You are required to create a Cas Consumer based on the Evaluator component of homework 2, and include it in your CPE pipeline.
3. (5 pts) Please name your CPE descriptor as hw3-ID-CPE.xml and put it under `src/main/resources/`, so that we could easily find the entry point of your pipeline.

Task 2

Deployment Architecture with UIMA-AS

In this task, you will need to learn what UIMA-AS is and how to use it. You are required to integrate a remote UIMA-AS service (Stanford CoreNLP) into your CPE pipeline, and deploy your aggregate analysis engine in homework 2 as an UIMA-AS service.

Task 2.1 Learning UIMA-AS

There are several useful resources to help you get started with UIMA-AS:

- You can learn basic concepts about UIMA-AS from *Getting Started: Apache UIMA Asynchronous Scaleout* (`http://uima.apache.org/doc-uimaas-what.html`).
- Once you downloaded UIMA AS binary package, you will find the README file (`http://svn.apache.org/viewvc/uima/uima-as/tags/uima-as-2.4.0/README?view=co`) contains practical guides for setup and running UIMA-AS tools.
- Also, you can view the detailed reference manual for UIMA-AS (`http://uima.apache.org/d/uima-as-2.4.0/uima_async_scaleout.html`).

Task 2.2 Creating an UIMA-AS client (25 pts)

Based on what you learned above, you need to create an UIMA-AS client descriptor (scnlp-ID-client.xml) for a remote UIMA-AS service (Stanford CoreNLP), and integrate your client with your CPE pipeline.

The UIMA-AS service we provided for this homework is the Stanford CoreNLP [1] Annotator from ClearTK toolkit [2]. This annotator reads the DocumentText from JCas

[1] `http://nlp.stanford.edu/software/corenlp.shtml`
[2] `https://code.google.com/p/cleartk/`

and do tokenization, sentence splitting (required by most Annotators), POS tagging, lemmatization, NER, syntactic parsing, and coreference resolution. The source code of this annotator is available to view online at `https://code.google.com/p/cleartk/source/browse/cleartk-stanford-corenlp/src/main/java/org/cleartk/stanford/StanfordCoreNLPAnnotator.java`. You may also be interested in ClearTK's type system `https://code.google.com/p/cleartk/source/browse/cleartk-type-system/src/main/resources/org/cleartk/TypeSystem.xml`.

To call and use this service, you need to import dependencies of cleartk-stanford-corenlp and uimaj-as-activemq to your maven project:

```
<dependency>
  <groupId>org.cleartk</groupId>
  <artifactId>cleartk-stanford-corenlp</artifactId>
  <version>0.8.0</version>
</dependency>
<dependency>
  <groupId>org.apache.uima</groupId>
  <artifactId>uimaj-as-activemq</artifactId>
  <version>2.4.0</version>
</dependency>
```

Finally, the UIMA-AS service for this homework has following metadata:

```
brokerURL: tcp://mu.lti.cs.cmu.edu:61616
endpoint: ScnlpQueue
```

You are required to integrate the **Name Entity** annotations from the Stanford CoreNLP service into your answer scoring component (15 pts), and compare the accuracy and the speed with your pipeline in homework 2 (10 pts).

Task 2.3 Deploying your own UIMA-AS service (40 pts)

In this task, you are going to deploy your aggregate analysis engine in Homework 2 on your own machine, and also call the service locally.

1. (10 pts) You need to first create a deployment descriptor (hw2-ID-aae-deploy.xml) for your aggregate analysis engine (hw2-ID-aae.xml);
2. (10 pts) Start a UIMA-AS broker locally, and deploy your service to your local broker;
3. (10 pts) Create a client descriptor (hw2-ID-aae-client.xml) for your service;
4. (10 pts) Create a CPE descriptor (hw3-ID-aae-as-CPE.xml) to test your service by calling the client.

Tips: If you are going to deploy your service with UIMA-AS command line tools, the maven plugins dependency:copy-dependencies or jar-with-dependencies might help you set up UIMA_CLASSPATH.

Task 2.4 Bonus (up to 20 pts)

There will be bonus points, if you can:

1. (5 pts) run a Stanford CoreNLP annotator locally, and compare the speed with the remote one;
2. (5 pts) incorporate other annotations from Stanford CoreNLP such as POS-tagging, lemma, and parsing;
3. (10 pts) create additional UIMA-AS service, that deploy other tools (e.g. Semantic role labeling) as service and can be called by your classmates. You can form a team of one or two (you can find you teammate on Piazza!) to deploy and test your tools between different machines. If you did this successfully, please let TAs know. We are glad to deploy your tools on our servers and share with the whole class.

Task 2.5 Writing up your report (10 pts)

We expect you to highlight the features of your design and your system, evaluation results, and comparison. Finally, don't forget to put your name and Andrew ID at the top of the document, name the file as "hw3-ID-report.pdf" and put it under `src/main/resources/docs`.