CSCI 578 – ASSIGNMENT 1

Deadline: by 9am on Wednesday, February 13

Submission Information: Please submit on D2L (details near the end of this document).

WHAT THIS ASSIGNMENT IS ABOUT

Now that you have a working platform to run software architecture recovery on, it's time to put it to use. In this assignment, you will recover architectural views of two versions of a software system. You will use three different architecture recovery techniques and compare the resulting architectural views as described below.

BEFORE YOU GET STARTED

This assignment requires you to have finished the previous assignment (0.5) successfully and still have access to your VM. If this was/is not the case, please notify us immediately (preferably via Piazza).

FIRST STEP – GETTING THE SYSTEM VERSIONS

You will need to download two compiled versions of the same system. ZIP archives containing individual compiled versions of several systems are available here.

Again, you will need to download TWO versions of the SAME system. Another requirement is that the first of these two versions has not been downloaded yet by anyone. This "first come, first served" requirement is to ensure that everyone has at least one unique version of a system. A list of the students who have already downloaded systems and their names is here. (You will need to sign on with your usc.edu credentials to add your information.)

Make sure you download a first version that nobody else has downloaded yet.

It is recommended that you download the software using Firefox, which is installed in your VM and can be started by clicking on the Firefox icon on the doc.

SECOND STEP – UNZIPPING THE DOWNLOADED FILES

In the RecProjects folder on your desktop, please create a folder that is named after the system you have downloaded (not its versions). For example, if you've downloaded chukwa-0.1.2.zip and chukwa-0.8.0.zip, then your system is called chukwa, and you should call your folder "chukwa".

Its directory structure should look something like this:

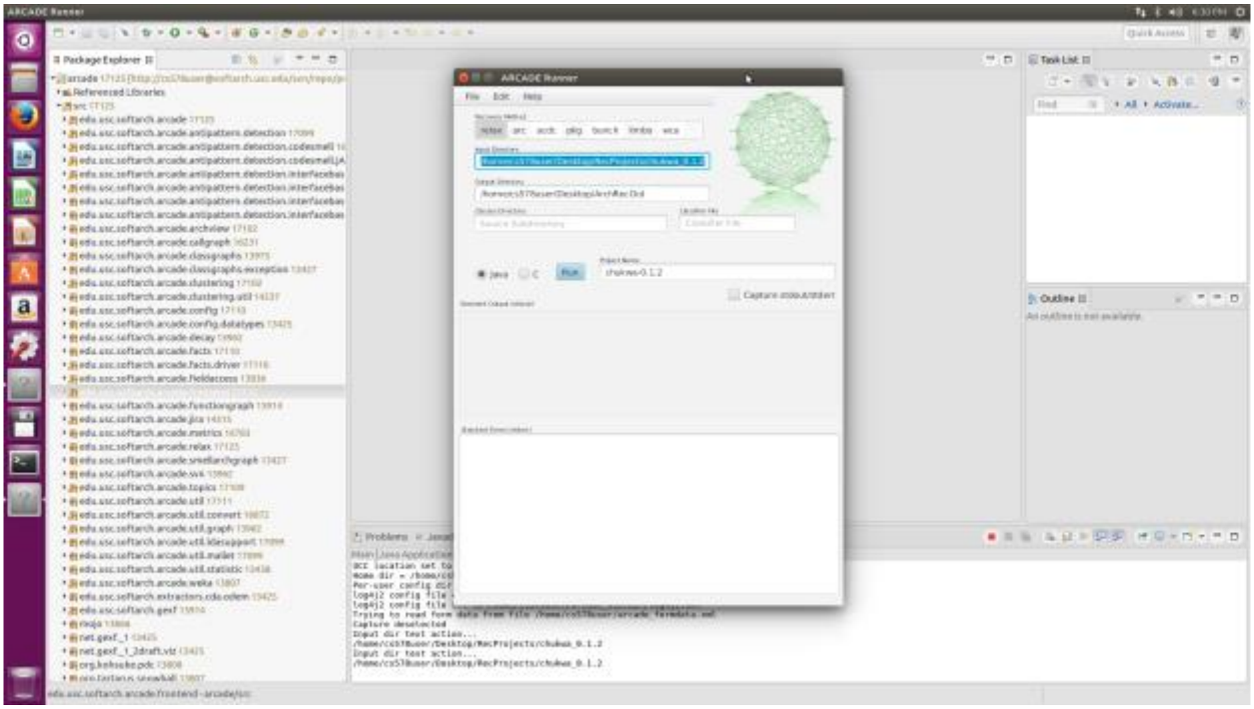
chukwa|

- chukwa-0.1.2
  - bin
  - build
  - …
- chukwa-0.8.0
  - bin
  - Changes.txt
  - …

Congratulations, you are now ready to recover!

THIRD STEP – RUNNING RECOVERIES

Just do the exact same thing you did in assignment 0.5, which should bring up this:



```
Trying to read form data from file /home/cs578user/arcade_formdata.xml
Capture deselected
Input dir text action...
/home/cs578user/Desktop/RecProjects/chukwa_0.1.2
Input dir text action...
/home/cs578user/Desktop/RecProjects/chukwa_0.1.2
```

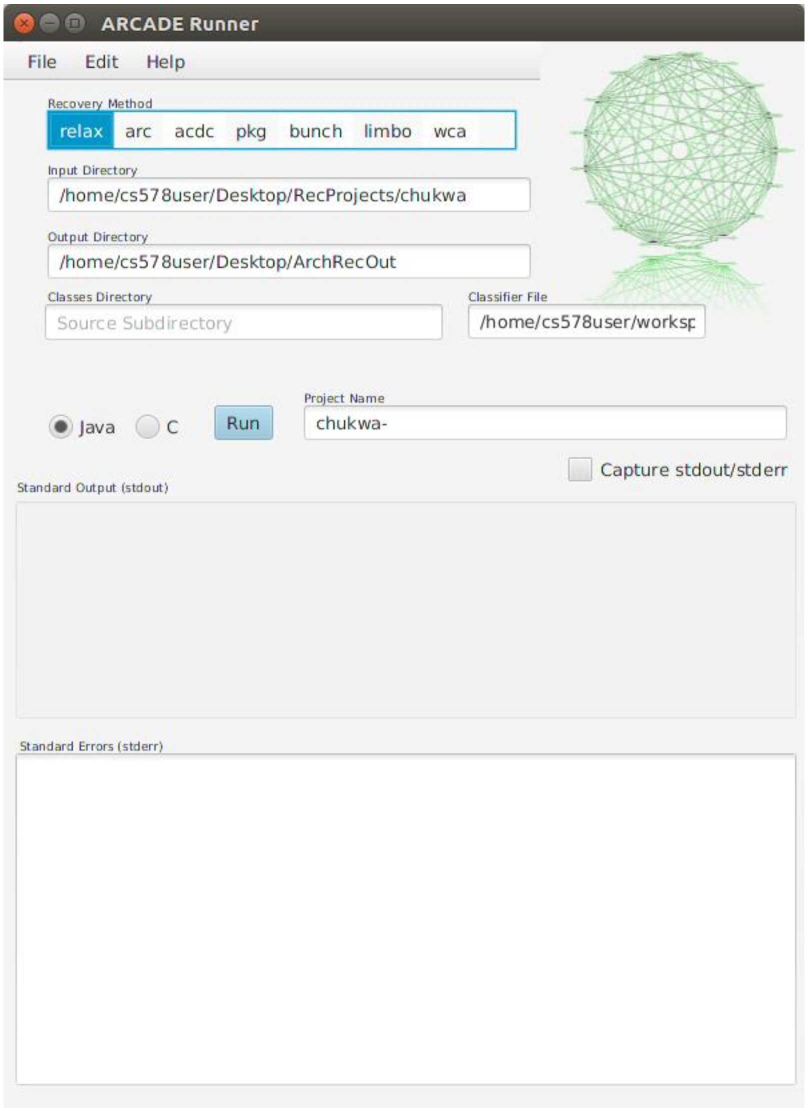
You will now need to enter the directories and parameters for your recoveries. All directory and file settings can be made by double-clicking on the text field for the file names. This will bring up a file selector.

Your input directory should be the one that concerns the system versions that you have unzipped in the previous steps. For our example, it would be /home/cs578user/Desktop/RecProjects/chukwa. Your output directory should be /home/cs578user/Desktop/ArchRecOut, which should already be set (but if it isn't, just select it again). You don't need to select a classes directory, so that field should stay empty. Your classifier file should be /home/cs578user/workspace/arcade/classifiers/relax.classifier.

As a language, you should select Java if it isn't already selected.

The selected recovery method for your first run should be "relax" (which should already be selected).

Overall, the ARCADE Runner dialog should look similar to this:



Recovery Method
relax arc acdc pkg bunch limbo wca

Input Directory
/home/cs578user/Desktop/RecProjects/chukwa

Output Directory
/home/cs578user/Desktop/ArchRecOut

Classes Directory
Source Subdirectory

Classifier File
/home/cs578user/worksp

● Java ○ C

Project Name
chukwa-

□ Capture stdout/stderr

Standard Output (stdout)

Standard Errors (stderr)

Now you are ready to launch the recovery!

Click on "Run".

ARCADE will now run the recovery.

The output will go into the ArchRecOut directory that was specified further above. For each recovery run, a new subdirectory will be created whose name is made up of the name of the system, the recovery method, the selected language and a time/date stamp. One example could be "chukwa-_RELAX_java_2017-01-27_22_31_29_123".

After the recovery run, you will find several files in that directory, as described in the lecture about ARCADE on January 11.

After the RELAX recovery, restart ARCADE (by exiting the ARCADE Runner dialog and restarting it). Your directory selections should all still be the same as before.

Now select ARC as a recovery method in the dialog and run it. Don't panic if this takes a while (possibly a few hours to finish – this is not a reason for concern. But do start early with your assignment so that you can spare the time needed for ARC to complete its analysis.

Lastly, select ACDC as a recovery method and run it.

FOURTH STEP – ANALYSIS AND WRITEUP

You will now have recovered two versions of the same system with three different recovery methods, which will have given you three architectural views of each version.

For each system, compare the architectural views of the pair of versions from that system to each other and create a write-up of at least 100 words for that comparison. In order to help you with this part of the assignment, Appendix A contains a description of the three recovery methods.

Aspects that are possible for you to cover include, but are not limited to:

- A summary of the most noticeable differences between the two versions according to the recovered architectural views.
- Why does this recovery method show these changes and not others?
- Do you think the differences accurately reflect the changes in the system that have taken place between the two versions?
- Based on the results of the recovery, what conclusions would you draw about the future of the system?

PLEASE NOTE

- When ARCADE starts up, you may get some console errors about the StatusLogger and the GUI. These are not of concern.
- You'll need to restart ARCADE after each recovery run by closing the ARCADE Runner dialog and restarting it. (This is because not all data structures may be reinitialized after the first run.
- If your screen resolution is different from the screenshots shown here, it will lead to variations in how things will look on your screen. This is no reason for concern.
- Make sure you are registered on our piazza.com class discussion forum. Instructions for registering have been posted to D2L in the announcements.
- If you encounter any issues with any of the steps discussed here, please check on Piazza whether your issue has already been addressed. If not, please post a question on Piazza.

DELIVERABLES

Please submit your files on D2L.

For each of the recovery runs, you'll need to submit different files.

1. From the RELAX run, please submit the two "…directories.pdf" files.
2. From the ARC run, please submit the two "…deps_clusters.pdf" files.
3. From the ACDC run, please submit the two "…acdc_clustered.rsf" files.

Additionally, you'll need to submit your write-up in PDF format.

GRADING

This assignment will constitute 10% of your grade. From a possible 100% for the assignment, you will receive points as follows:

- 5% for each valid output files listed in the "Deliverables" section,
- up to 70% for the write-up.

(*) Files that are considered valid are those that contain output from ARCADE runs.

APPENDIX A – DESCRIPTIONS OF THE THREE RECOVERY METHODS

All three algorithms operate on the source code of a system and how it's organized in directories. That is, they're basing their architectural views on the system at rest, as opposed to looking at it while it's running.

ACDC clusters source files by following patterns that commonly occur in manual decompositions (manual recoveries) of software systems. It tries to identify subsystems using a collection of criteria that are based on connections between entities (such as directory structure, headers and body, support libraries and others), names the subsystems and then creates clusters from them.

ARC applies topic modeling, an NLP technique, to the source code. It treats the source code of a given system version as a body of text and determines which groups of words occur together most frequently. These groups of words are called "topics". For each source file it then determines which topic it is most aligned with and groups the source files into clusters for that topic. Parameters that need to be set for each run are the number of topics and the number of clusters. The meaning of a topic that has been found by ARC can only be determined by humans. (Note that we will not ask you to set any parameters or to determine the meaning of topics in this assignment. The parameters are already pre-set for you.)

RELAX applies text classification, another NLP technique, to the source code. It looks for predefined topics in each source file and determines which topic it is most aligned with. It then groups the source files into clusters for each topic. For it to be able to run, a classifier has to be trained from a set of groups of documents where the documents in each individual group are all related to one topic. No parameters have to be set for each individual run. (Note that we will not ask you to train any classifiers – those will be provided to you by us.)