Summary for the feedback from Chandan in Design Review for Milestone 2

The design is effective to make the project support the features in Milestone 2 without modifying lots of existing code. It is reasonable to separate the four types of dependency into parts to be handled in different transformers. Adding the new SDRenderingTransformer is the right place to implement the rendering of sequence diagram. That way we can also reuse our PlantUmlMaker to generate the svg of the uml. To shorten the SDRenderingTransformer in terms of the lines of code needed, a new class can be considered to be created to find the SootMethods and return back to the SDRenderingTransformer. Thus all the SDRenderingTransformer does is take the SootClass and recursively look up the classes it invokes using callGraph. Using callGraph can prevent us from creating and tracking another data structure similar to the existing Dependency class. The advice on using callGraph is that choose the first option when finding the target of method calls. Lastly, the requirement of maximum call depth of 5 excludes Java classes.