Sprint 2 Design and Project Plan

Kennedy Bombei, Ayman Noreldaim, Dylan Fair, and Sam Burkhart

4/21/2023

Introduction

Overview:

The goal of this sprint was to have a mostly complete and functional ReliabilityVisualization class that creates an .ra file that evaluates the end-to-end reliability of the warp flows if the –ra flag is requested. The overall goal by the end of the project is to have complete and functional ReliabilityVisualization and ReliabilityAnalysis classes to create the appropriate .ra file located in the output files.

Recap of Sprint 1:

Dylan and Sam created the Sequence Diagram showing program flow starting with Warp processing the –ra option. Kennedy and Ayman developed the project plan and updated the ReadMe file.

Sprint 2: Dylan, Kennedy, and Sam developed most of the code to create the .ra file and fill it with the appropriate number of columns and rows. We created a method, to create and fill the header with the flow name, min package reception rate, e2e reliability, and the number of channels. The method numColumns calculates the number of columns for our .ra file, with a column for every node in every flow. We have another method in this class that creates the name for each column by combining the flow with each node. We also created a createVizualizationData method that is supposed to fill our .ra file with the end-to-end reliabilities of the warp flows. We are currently calculating this using the equation: newSinkNodeState = (1-M)*prevSnkNodeState + M*prevSrcNodeState. This currently does not fill the .ra file with the correct information, but we will be fixing the output in Sprint 3. Ayman wrote the JUnit tests for our reliabilityVizualization class, mostly using the Example.txt output given to compare values. Since our createVizualizationData method does not return the correct output, some of these tests fail but they will test for the correct output, and when we fix this method in Sprint3 all of the tests will pass. Kennedy completed this Design and Project Plan Document and updated the ReadMe file.

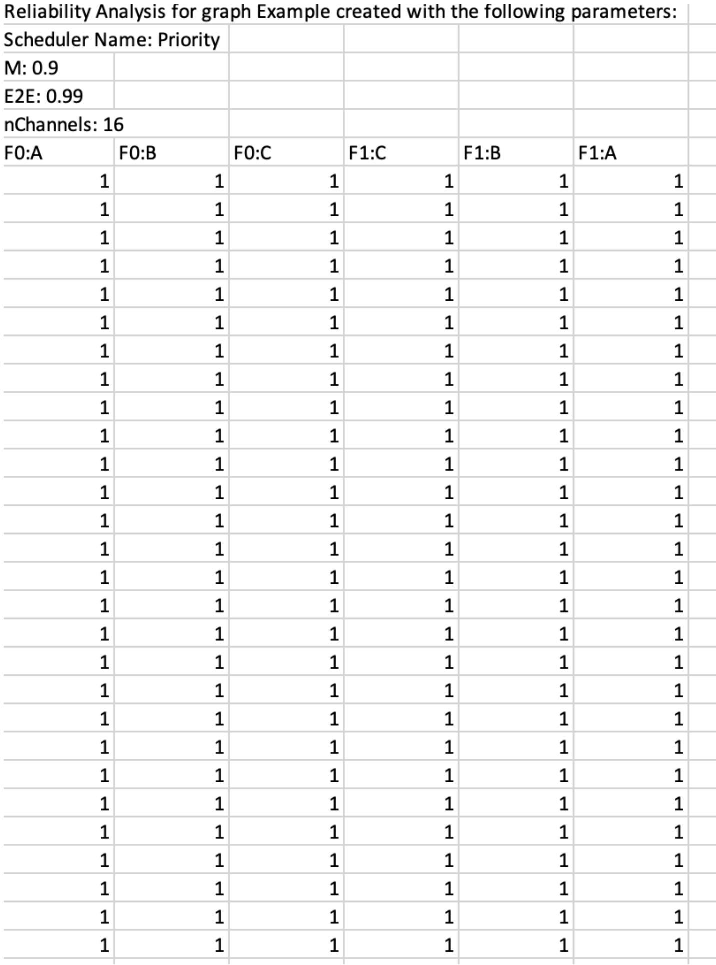
Sprint 3 Plan (Future Goals):

In Sprint 3 we will change the createVisualizationData method to give the correct data output for our .ra file. We plan to do this by calling methods from the reliability analysis class. We need to correct the implementation of our getReliabilities method to return the correct data. We also will use InstructionParameters class to help populate the data for our .ra file. Specifically ,we will utilize this method call : ArrayList<InstructionParameters>getInstructionParameter(String instruction). Similar to Sprint 2, Dylan, Kennedy, and Sam will focus on the program design and code correctness. Kennedy is going to update the JavaDoc comments and UML diagram as we create and change methods, Sam is going to update the ReadMe files, and Dylan will take the lead on writing the code with input and contributions from all other group members. Ayman will be taking the lead on the JUnit tests, adding on to the tests he previously created for Sprint 2, with input and contributions from all other group members.

Solutions

Our Current "Solution":

Currently our code successfully generates the .ra file below (when ran with the Example.txt file), which is generally formatted correctly, but does not contain the correct data values in each cell. Our header and all elements contained in it are correct, as are the dimensions of our visualization graph.



Reliability Analysis for graph Example created with the following parameters:					
Scheduler Name: Priority					
M: 0.9					
E2E: 0.99					
nChannels: 16					
F0:A	F0:B	F0:C	F1:C	F1:B	F1:A
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

*There are more columns than shown in this image. *

Our proposed solution (described in the Sprint 3 section) will return a similar output file but instead of 1's in every cell the correct data will be calculated.

Testing Plan:

For our JUnit Tests in Sprint 2, we are going to test by reading the .ra file generated by ReliabilityVisualization class and we are testing the individual methods we have created. For the methods we have generating the correct output we will have passing tests, however if we compare the data in our .ra file to the Example.txt, that test would not pass at this stage. In Sprint 3, after correcting the VisualizationData method, we will directly compare this method and the generated .ra file to the expected values from Example.txt and other files. We will also test the getReliabilites method in Sprint 3 once we correctly implement it, in addition to any other methods or getters/setters that we create in the process. We will rerun any old tests from Sprint2 to ensure that our changes don't break anything else in the code.

Success Evaluation

Overall, we succeeded in Sprint 2 as we developed methods that create and populate a .ra file based on the end-to-end reliability of the warp flows. Our code runs with no errors and generates a file that is almost accurate, which indicates good progress has been made. Ideally our file would have the correct data, but by correcting our CreateVisualization method as described in the above plan for Sprint 3, we will be right on track to complete the project in the timeframe given for Sprint 3.

Work

Work Estimates and Timeline:

For Sprint 2, we met during our discussion section each week and worked on the project, and in addition we met in person outside of class and had multiple zoom meetings to discuss our plan and work on the project. For Sprint 3, we predict a similar amount of time will be needed. We have a goal of completing Sprint 3 multiple days before the due date of May 5th, so that we can use the extra time to clean up our code and documentation and to ensure that all project requirements have been met.

Prioritization:

For Sprint 2, our priority was to create an output .ra file with the correct header and a visualization with the correct dimensions for our data. We also prioritized testing since this is a big portion of the grade for this assignment. We met this goal multiple days before the assignment deadline and were able to use the extra time to finish documentation, update all diagrams used, and started working on Sprint 3 (generating the correct data ta output). For Sprint 3 our priority is going to be ensuring that our design and code are correct and efficient, with our second priority on JUnit testing.