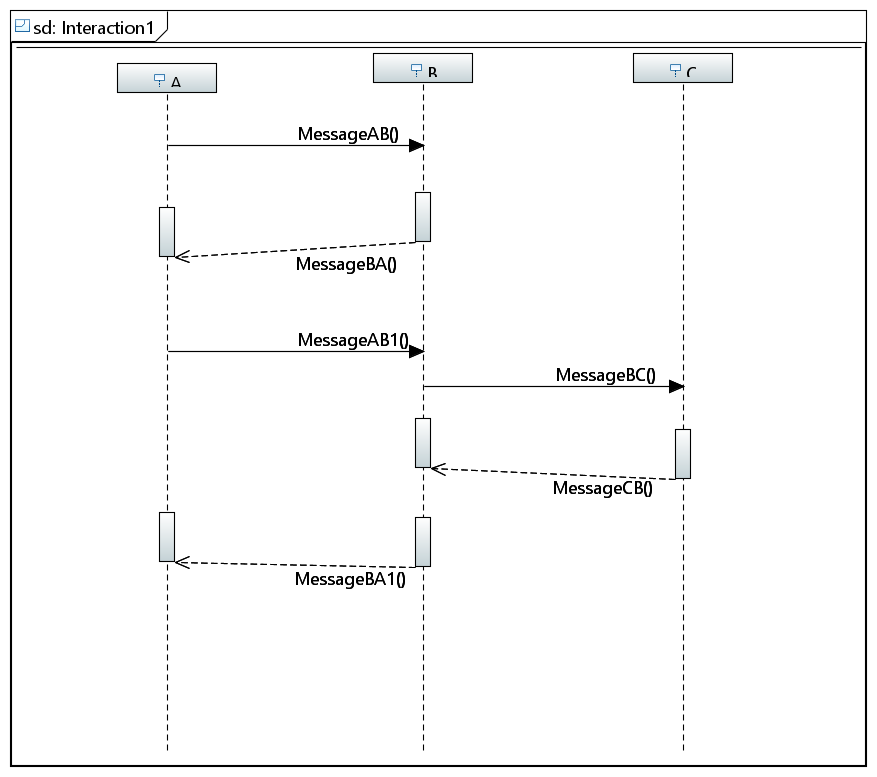
**TEST CASE GEN Version-1 INSTRUCTIONS/USER MANUAL**

**(12/27/2016 -Nephi, Nikhil)**

1. Install Papyrus: <https://eclipse.org/papyrus/download.html>
2. Test Case Gen:
   1. Clone/Download Repository: <https://github.com/nikhil1710/MasterProject.git>
   2. Extract zip:
      1. In Eclipse select File -> Open Projects from File System
      2. In the source field select directory and browse through “MasterProject-master” -> SEProject and select this as the root directory
      3. Click finish
      4. The project will be added to eclipse workspace and needs required libraries in the Java build path to run
      5. In the project explorer window right click -> properties -> java build path -> remove all the libraries that are not properly referenced -> Add external jars -> browse through “MasterProject-master” -> SEProject -> src -> Libraries – TCG -> add all libraries in this folder -> click ok
      6. Select Testing.java and click ctrl+f11 which should produce the output in an applet window
      7. The target test code is created in sub folder called target
3. Create UML Interaction Diagram in Papyrus

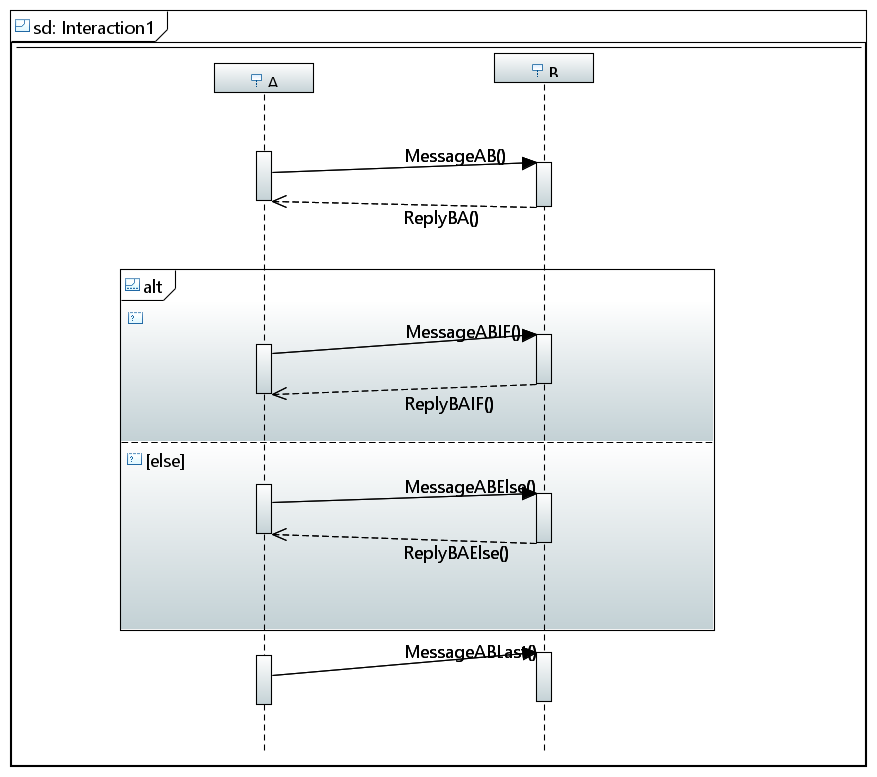
The following are the sample UML diagrams created in Papyrus for testing this tool:

1. Simple case:
   1. Diagram



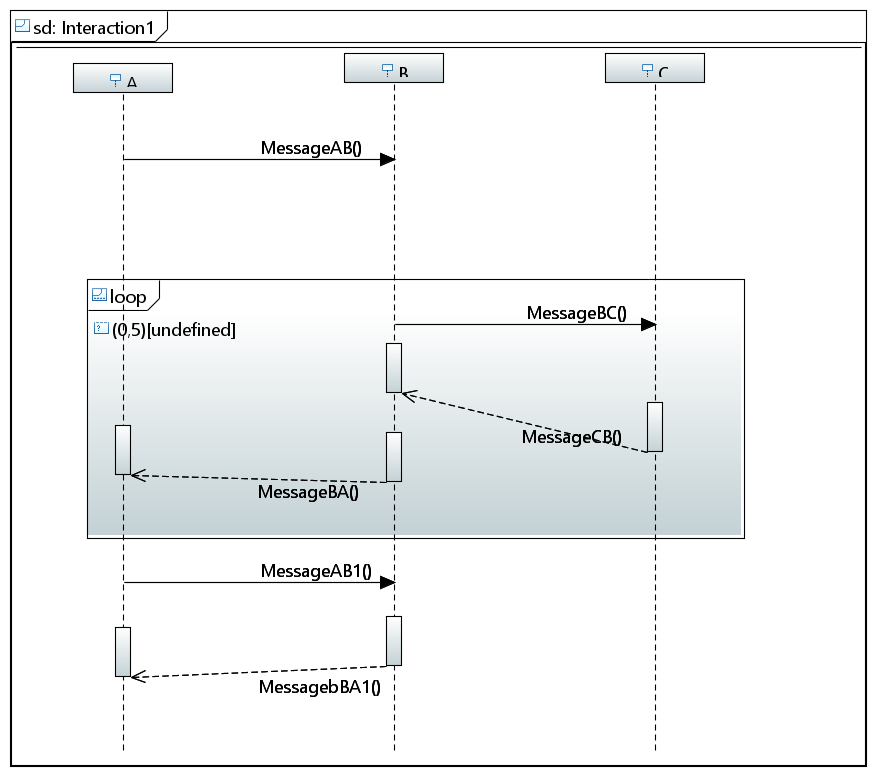
* 1. XML LINK: <https://github.com/nikhil1710/MasterProject/blob/master/SEProject/src/UMLInput/Cases/Simple%20case/Simplecase.uml>

1. Alt Case:
   1. Diagram:



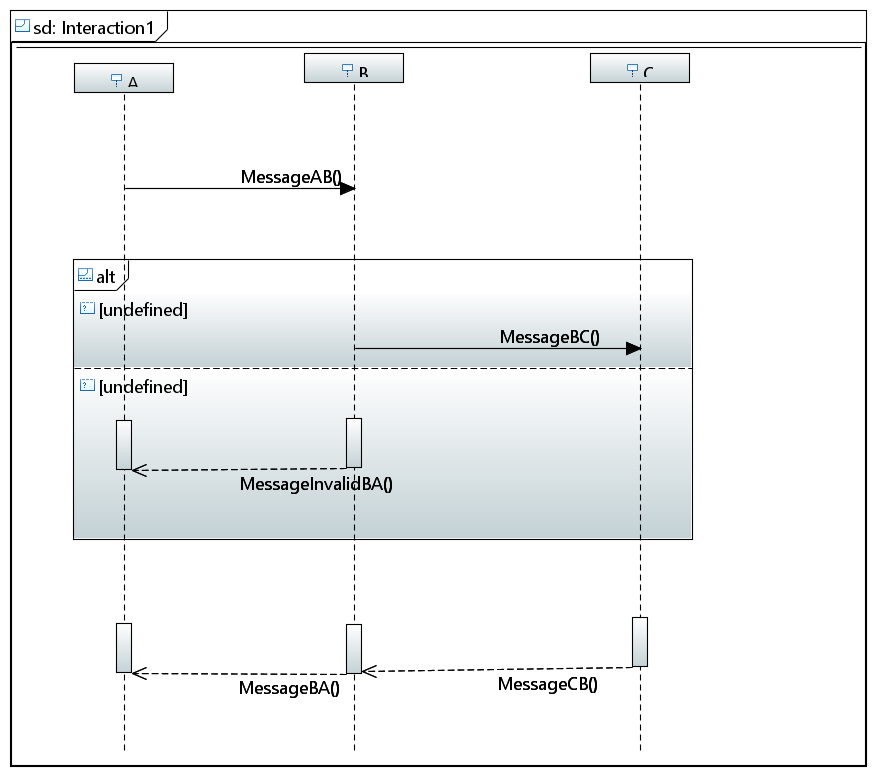
* 1. XML Link: <https://github.com/nikhil1710/MasterProject/blob/master/SEProject/src/UMLInput/Cases/Alt%20case/ALT.uml>

1. Loop Case:
   1. Diagram:



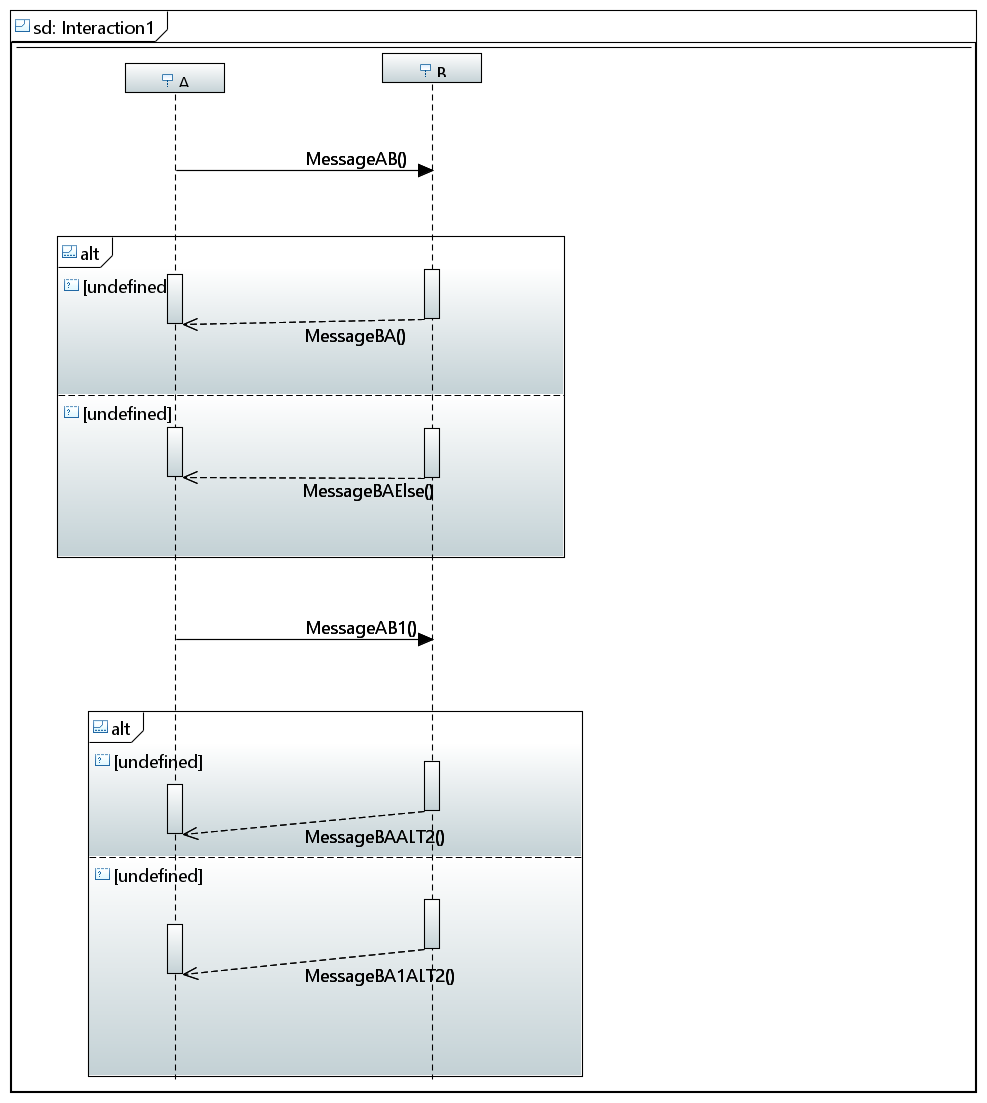
* 1. XML Link: <https://github.com/nikhil1710/MasterProject/blob/master/SEProject/src/UMLInput/Cases/Loop%20case/Loop.uml>

1. Alt with 3 lifelines case:
   1. Diagram:



* 1. XML Link: <https://github.com/nikhil1710/MasterProject/blob/master/SEProject/src/UMLInput/Cases/Alt%20with%203%20Lifelines/FinalModel.uml>

1. Two Alts case:
   1. Diagram:



* 1. XML: <https://github.com/nikhil1710/MasterProject/blob/master/SEProject/src/UMLInput/Cases/Two%20Alts/twoAlt.uml>

1. The tool even works for loop followed by an alt
2. Copy the XML generated in Papyrus project to ECLIPSE UML input folder
3. Run the Program. Graph gets generated in a separate window while test code gets created in the target folder of the project.

\***LIBRARIES USED:**

1. CODEMODEL – FOR AUTO CODE GENERATION
2. GRAPH STREAM -TO DISPLAY TEST CASES IN GRAPHICAL VIEW

**CODE ASSUMPTIONS:**

1. Reads input(.uml/.xml) from UMLInput folder. Test with one of the XML samples provided above.
2. Code doesn't work for a UML diagram with nested 'loops' and 'alt' boxes.
3. Test Case Gen works for only Papyrus UML tool.

**BREAKING DOWN CODE:**

* We initially read the input file in the main functions. Used DocumentParser to parse the entire XML file.
* We store all the required information in HashMaps. We use 3 functions in order to do this,
  1. **generateFragments**: Stores all the Fragment information from XML.
  2. **generateMessageMap:** Stores the message information. (Key: MessageID, Value: MessageName).
  3. **generateLifeLineMap:** Stores all the lifeline information. (Key: LifeLineID, Value: LifeLineName).
* We then used 3 different functions which ultimately generates all the test cases.
  1. **generateScenarios:** Generates all possible scenarios(paths) in the system. This is stored in the form of a HashMap.
  2. **generateTestSuite:** Generates all the test cases(Class and its respective methods) from the generated scenarios.
  3. **generateGraph:** Generates the graph on a separate window showing all possible paths in the system.
* We also used DeepClone class, to clone and copy our HashMaps.
* Keep in mind that we used many delimiter's in our HashMaps, to differentiate values.

**CONTACT**:

**Nikhil Reddy Sudireddy: sudireddy.ne@husky.neu.edu**

**Nephi Calvin Bandela: bandela.n@husky.neu.edu**