Test Plan

For agentTool III (Dynamic)

Version 1.0

Submitted in partial fulfillment of the requirements of the degree of MSE

Binti Sepaha
CIS 895 – MSE Project
Kansas State University
TABLE OF CONTENTS

1. Test Plan Identifier ................................................................................................................................. 4

2. Introduction ............................................................................................................................................ 4

3. Test Items ............................................................................................................................................... 4

4. Features to be tested ............................................................................................................................... 4

5. Features not to be tested ......................................................................................................................... 5

6. Approach ................................................................................................................................................ 5

7. Item Pass/Fail Criteria ........................................................................................................................... 6

8. Suspension Criteria and Resumption Requirements .............................................................................. 6

   8.1. Suspension Criteria....................................................................................................................... 6

   8.2. Resumption Requirement ............................................................................................................. 6

9. Test Deliverables.................................................................................................................................... 6

   9.1 Test Log......................................................................................................................................... 6

10. Testing Tasks.......................................................................................................................................... 6

    10.1. Test Case 1 – SR1.1 - SR1.4 – Sequence Diagrams................................................................. 6

    10.2. Test Case 2 – SR1.5 - SR1.16 ..................................................................................................... 7

    10.3. Test Case 3 – SR1.17 - SR1.19 .................................................................................................. 7

    10.4. Test Case 4 – SR1.20 - SR1.22 ................................................................................................... 7

    10.5. Test Case 5 – SR1.24 - SR1.25 ................................................................................................... 8

    10.6. Test Case 6 – SR2.1 – SR2.3 – Activity Diagrams ................................................................. 8

    10.7. Test Case 7 – SR2.4 – SR2.9 ...................................................................................................... 8
10.8.   Test Case 8 – SR2.10 – SR2.17................................................................................................... 9

10.9.   Test Case 9 – SR2.18 – SR2.19................................................................................................... 9

10.10. Test Case 10 – SR2.20 – SR2.25............................................................................................... 9

10.11. Test Case 11 – SR2.26 – SR2.29................................................................................................ 10

10.12. Test Case 12 – SR2.30 – SR2.32............................................................................................... 10


1. **Test Plan Identifier**
   agentTool-Dynamic-V1.0

2. **Introduction**
   This document describes the methods that will be used to test the entire functionalities of agentTool III (Dynamic). The tool allows the user to draw behavioral diagrams like Sequence and Activity Diagrams. Each diagram will be treated as a separate module of the system. Each module will be tested with respect to the requirements as described in the vision document.

3. **Test Items**
   The following core system modules will be tested
   - Drawing Sequence Diagrams
   - Drawing Activity Diagrams
   - Importing XML Models
   - Exporting XML Models
   - View System
   - Printing Diagrams

4. **Features to be tested**
   The following list of features will be tested for each diagram in agentTool III (Dynamic). The features reference the Specific Requirements (SR) outlined in the Vision document.
   
   **Drawing Sequence Diagrams (SR1)**
   Features mentioned in Vision Document 1.0 section 3.1.1.1 starting from SR1.1 to SR1.25

   **Drawing Activity Diagrams (SR2)**
   Features mentioned in Vision Document 1.0 section 3.1.1.2 starting from SR2.1 to SR2.32
   SR3.23 System Object Model Generation - Vision Document 1.0 section 3.1.3

   **Importing XML Models (SR4)**
   SR4.1 Loading Diagrams from an XML file - Vision Document 1.0 section 3.2.4.1

   **Exporting XML Models (SR5)**
   SR5.1 The system model will be saved to a file - Vision Document 1.0 section 3.3.3
SR5.2 The file will be in an XML compatible format - Vision Document 1.0 section 3.3.3

**View System (SR6)**

SR6.1 The user will be able to click on any diagram in the hierarchy to view or edit it - Vision Document 1.0 section 3.4.3

SR6.2 The user can delete diagrams or create new ones as part of the loaded system - Vision Document 1.0 section 3.4.3

**Printing Diagrams (SR7)**

SR7.1 Printing Diagrams on paper – Vision Document 1.0 section 3.5.3.1

SR7.2 Select the printer - Vision Document 1.0 section 3.5.3.2

SR7.3 Scale the diagrams to fit the paper - Vision Document 1.0 section 3.5.3.3

5. **Features not to be tested**

The following future requirements will not be tested.

**Drawing State Chart Diagrams (SR3)**

Features mentioned in Vision Document 1.0 section 3.1.1.3 starting from SR3.1 to SR3.22

SR1.23 Message overlapping will be represented by making a bridge between the messages.

SR1.26 Add timing constraints between two messages - Vision Document 1.0 section 3.1.1.1

SR1.27 Remove timing constraints - Vision Document 1.0 section 3.1.1.1

6. **Approach**

Only functional black box testing will be performed to test the functionality of agentTool III (Dynamic). The features describe how the user will interact with the system, so the testing will require the tester to interact with the system in the same way a typical user would. To simulate a typical user’s actions a set of scenarios will be created which describe a set of actions to take in order to achieve a desired result. Each scenario will trace back to a requirement listed in the Vision Document 1.0. However, in order to test that the object model is being correctly maintained, debug statements will be entered into the code and debug logs will be inspected.
7. **Item Pass/Fail Criteria**
   Test cases executed on agentToolIII (Dynamic) will pass if they meet the specific requirements as mentioned in the Vision Document. A test case will fail if any behavioral expectation is not met as described.

8. **Suspension Criteria and Resumption Requirements**
   8.1. **Suspension Criteria**
      If a test case fails, testing will be suspended for all dependent features. The failed test case will be logged into a test log along with a description of the failure.

   8.2. **Resumption Requirement**
      Test cases not dependent on the case in which a bug is reported will continue to be executed in parallel to bug fixing. Testing for the failed test case will resume after the bug has been identified and resolved.

9. **Test Deliverables**
   9.1. **Test Log**
      The Test Log will document all test cases and record if the test case passed or failed. A test case that fails will have the reasons for failure and suggested solutions documented.

10. **Testing Tasks**
    10.1. **Test Case 1 – SR1.1 - SR1.4 – Sequence Diagrams**
      **Tasks**
      - Click on the Sequence Diagram option on the tree hierarchy.
      - Resize the frame
      - Delete the frame
      - Edit the protocol name
      - Add/Modify parameters to the protocol
      **Results expected**
      - A new sequence diagram panel should appear with a default frame and a protocol name.
      - The user should be able to resize the frame
      - The frame cannot be deleted
      - The user should be able to modify the protocol name
      - Parameters should be added and modified as required by the user
10.2. Test Case 2 – SR1.5 - SR1.16

Tasks

• To an existing sequence diagram, add an alternative action frame from the palette
• Edit its name
• Move it within the sequence diagram frame
• Delete this frame.
• Repeat the same process for the ‘action in loop’ and ‘reference to another protocol’

Results expected

• A new frame for alternative action should be added within the existing sequence diagram frame.
• The user should be able to edit the name
• The user should be able to move this frame inside the existing sequence diagram
• The frame should be deleted
• The same results are expected for ‘action in loop’ and ‘reference to another protocol’

10.3. Test Case 3 – SR1.17 - SR1.19

Tasks

• Click on the class role icon on the palette and drag it on the sequence diagram panel.
• Delete/Edit these class roles.

Results expected

• A new class role should be added to the sequence diagram frame
• The user should be able to edit and delete it too
• The life line should be attached to it when the user drags it.

10.4. Test Case 4 – SR1.20 - SR1.22

Tasks

• Select message icon from the palette and then click on lifelines of two class roles (same or different)
• Add/edit label to the message
• Delete this message

Results expected

• A message should be created from the first class role to the second
• The user should be able to add/edit label to the message
• The message should be deleted

10.5. Test Case 5 – SR1.24 - SR1.25

Tasks
• Select stop icon from the palette and then click on a lifeline of a class role
• Delete this stop sign.

Results expected
• The life line should be marked with a ‘X’ sign which denotes that this class role no longer interacts in the protocol
• The stop sign should be removed

10.6. Test Case 6 – SR2.1 – SR2.3 – Activity Diagrams

Tasks
• Click on the Activity Diagram option on the tree hierarchy.
• Resize the frame
• Add Swim lanes from the palette to this frame
• Associate a role name with the swim lane
• Delete the frame

Results expected
• A new activity diagram panel should appear with a default frame and a protocol name.
• The user should be able to resize the frame
• Swim lanes should be added to the frame
• A role name should be associated with the swim lane
• The frame cannot be deleted

10.7. Test Case 7 – SR2.4 – SR2.9

Tasks
• To an existing activity diagram, add an initial/final node from the palette
• Add another initial/final node
• Send a message from this node to an activity
• Send a message to this node from an activity
• Delete this node.

Results expected
• An initial/final node should be added to the existing activity diagram frame
• The user should not be able to another similar node
• The user should be able to only send a message from an initial node and not from a final node
• The user should not be able to send a message to an initial node but should be able to send one to a final node.
• The added node should be deleted

10.8. Test Case 8 – SR2.10 – SR2.17

Tasks
• To an existing activity diagram, add an action state from the palette
• Add/Edit its name
• Add an action flow from the palette between two action states
• Delete this action flow
• Delete this action state

Results expected
• A new action state should be added to the existing activity diagram frame.
• The user should be able to add/edit the name
• An action flow should be added between two action states
• The action should be deleted
• The action state should be deleted

10.9. Test Case 9 – SR2.18 – SR2.19

Tasks
• To an existing activity diagram, add a synchronization point from the palette
• Add action flows to and from the synchronization point to action states
• Delete it.

Results expected
• A synchronization point should be added to the existing activity diagram frame
• Action flows should be added to and from the synchronization points to action states
• The synchronization point should be deleted

10.10. Test Case 10 – SR2.20 – SR2.25

Tasks
• To an existing activity diagram, add an outgoing/incoming event from the palette
• Add action flows to and from the added event
• Delete the added event.

Results expected
• An outgoing/incoming event should be added to the existing activity diagram frame.
• Action flows can only be sent by incoming events and received by outgoing events and not vice-versa
• The added event should be deleted

10.11. Test Case 11 – SR2.26 – SR2.29

Tasks
• To an existing activity diagram, add a decision node from the palette
• Add/edit guards to the outgoing action flows
• Delete the decision node

Results expected
• A decision node should be added to the existing activity diagram frame
• The user should be able to add/edit the guards on the outgoing action flows
• The decision node should be deleted

10.12. Test Case 12 – SR2.30 – SR2.32

Tasks
• To an existing activity diagram, add a flow final node from the palette
• Add action flows to and from this node
• Delete the node.

Results expected
• A flow final node should be added to the activity diagram.
• Action flows should not be sent from a flow final node but only to it.
• The flow final node should be deleted


Tasks
• Bring up each diagram one by one.
• Drag and drop palette components onto the canvas
• Inspect the debug logs being generated when components are dragged and dropped onto the canvas
Results expected
• The debug logs should contain print statements detailing the changes to the object model being performed as components are dragged / dropped and changed.

Tasks
• Save a diagram by selecting the save option from the menu
• Inspect the generated XML file
• Reload the file into the tool
Results expected
• The tool should generate the XML model.
• The diagrams should reappear on loading the file as they were when save was initiated.

10.15. Test Case 15 – SR6.1 – SR6.2 – View System
Tasks
• Bring up a diagram by clicking on it in the tree hierarchy
• Create a new diagram by right clicking on the tree root
• Delete an existing diagram by right clicking on it
Results expected
• The selected diagram should appear on the canvas
• The tester should be able to create a new diagram. An empty canvas should appear for drawing the diagram
• The deleted diagram should no longer be visible in the hierarchy

10.16. Test Case 16 – SR7.1 – SR7.2 – Printing Diagrams
Tasks
• Bring up a diagram by clicking on the tree hierarchy
• Select the print button from the toolbar
• Select the required printer and then press OK
Results expected
• A printer dialog box should appear
• The printout should come out at the selected printer