Test Plan

For agentTool III (Static)

Version 1.0

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

Deepti Gupta
CIS 895 – MSE Project
Kansas State University
# TABLE OF CONTENTS

1. Test Plan Identifier .......................................................................................................................... 3
2. Introduction .......................................................................................................................................... 3
3. Test Items .............................................................................................................................................. 3
4. Features to be tested ............................................................................................................................ 3
5. Features not to be tested ....................................................................................................................... 5
6. Approach ............................................................................................................................................. 5
7. Item Pass/Fail Criteria .......................................................................................................................... 5
8. Suspension Criteria and Resumption Requirements ........................................................................... 5
9. Test Deliverables ................................................................................................................................ 5
10. Testing Tasks ...................................................................................................................................... 6

10.1 Test Case 1 – SR1.1 to SR1.3 ........................................................................................................ 6
10.2 Test Case 2 – SR2.1 – SR2.3 ......................................................................................................... 6
10.3 Test Case 3 – Goal Mode ............................................................................................................... 6
10.4 Test Case 4 – Organizational Model ............................................................................................ 6
10.5 Test Case 5 – Role Model ............................................................................................................. 6
10.6 Test Case 6 – Agent Model ........................................................................................................... 6
10.7 Test Case 7 – XML generation SR8 – SR9 ................................................................................ 6
10.8 Test Case 8 – Change View SR7.1 – SR7.2 ............................................................................... 6
10.9 Test Case 9 – Maintain object model ........................................................................................... 6
1. **Test Plan Identifier**  
   agentTool-Static-V1.0

2. **Introduction**  
   This document describes the methods that will be used to test the agentToolIII (Static). The tool allows a designer to draw four kinds of static diagrams – Goal Model, Role Model, Agent Model and Organizational Model. 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  
   - Goal Model  
   - Role Model  
   - Agent Model  
   - Organizational Model

4. **Features to be tested**  
   The following list of features will be tested for each diagram in agentTool III (Static). These features can be referenced in the Vision document.  
   **All diagrams**  
   **SR1.1** The user will be able to click on any diagram in the hierarchy to view or edit it.  
   **SR1.2** The user can delete a diagram  
   **SR1.3** The user can create new diagrams as part of the loaded system.  
   **SR2.1** Each model entity in a diagram will have attributes name and description which the user can edit  
   **SR2.2** Each diagram will provide a drawing capability in which the user can drag/drop the required icons onto the drawing space.  
   **SR2.3** The user will also be provided with the facility to remove/delete model entities, move entities around in the drawing space and modify their properties.  
   **SR2.4** Corresponding to each user action for every diagram, the system object model will be maintained by the tool in the background.  
   **Goal Model**  
   **SR3.2** AND Connectors
SR3.3 OR Connectors
SR3.4 Goal Auto Numbering
SR3.7 Goal Tree validation

Organization Model
The following icons will be provided to the user for drawing this diagram
   SR4.1 Organizations
   SR4.2 Goals
   SR4.3 Services
   SR4.4 Actors
   SR4.5 Protocols
   SR4.6 <<achieves>>
   SR4.7 <<provides>>

Role Model
The following icons will be provided to the user for drawing this diagram
   SR5.1 Goals
   SR5.2 Capabilities
   SR5.3 Services
   SR5.4 Actors
   SR5.5 Protocols
   SR5.6 Roles
   SR5.7 <<achieves>>
   SR5.8 <<requires>>
   SR5.9 <<provides>>

Agent Model
The following icons will be provided to the user for drawing this diagram
   SR6.1 Organizations
   SR6.2 Agents
   SR6.3 Services
   SR6.4 Actors
   SR6.5 Protocols
   SR6.6 Roles
SR6.7 Capabilities
SR6.8 <<plays>>
SR6.8 <<possesses>>

SR7.1 – SR7.2 Change View

5. Features not to be tested
   The following future requirements will not be tested
   SR3.5 Precedes relation between goals
   SR3.6 Goal Parameters

6. Approach
   Only functional black box testing will be performed to test the functionality of agentToolIII. The user actions will be simulated through a set of test scenarios. Each scenario will trace back to a requirement listed in the Vision document. 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 will pass if they meet the 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
       Testing for the failed test case will resume after the bug has been identified and resolved. Independent test cases will continue to be executed in parallel to bug fixing.

9. Test Deliverables
   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 to SR1.3

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

10.2. Test Case 2 – SR2.1 – SR2.3

For every component entity (i.e. components and links connecting them) from any of the four diagrams perform the following tasks

Tasks
- Drag and drop palette components onto the canvas
- Delete component entities
- Rename / change description for entities
- Connect the various components with link connectors

Results
- Each component can be dragged and dropped onto the canvas
- Each component can be deleted and edited successfully
- The relationship connectors must connect valid components. No invalid connections should be permitted. Valid connections and relationships are detailed in the Vision Document
- When the components are dragged around on the canvas, the connectors will also move around and remain connected appropriately.

10.3. Test Case 3 – Goal Model

Tasks
- Bring up a goal diagram
- Drag and drop palette components onto the canvas
- Draw invalid goal tree
- Check auto-numbering of goals
Results expected

- The drawing palette should contain all components required to draw a goal diagram
- The user should not be permitted to draw an invalid tree. No cycles allowed.
- Goals should be auto-numbered by the tool as they are connected to other goals.

10.4. Test Case 4 – Organizational Model

Tasks

- Bring up an organizational diagram
- Drag and drop palette components onto the canvas

Results expected

- The drawing palette should contain all components required to draw an Organizational diagram

10.5. Test Case 5 – Role Model

Tasks

- Bring up a role diagram
- Drag and drop palette components onto the canvas

Results expected

- The drawing palette should contain all components required to draw a role diagram

10.6. Test Case 6 – Agent Model

Tasks

- Bring up an agent diagram
- Drag and drop palette components onto the canvas

Results expected

- The drawing palette should contain all components required to draw an agent diagram

10.7. Test Case 7 – XML generation SR8 – SR9

Tasks

- Save a diagram by selecting the save option from the menu
- Inspect the generated XML file
- Reload the XML file into the tool

Results expected

- The tool should generate XML in accordance to the DTD specification developed
• The diagrams should reappear on loading the XML file as they were when save was initiated. The goal diagram should be visible on the canvas by default

10.8. Test Case 8 – Change View SR7.1 – SR7.2

Tasks
• Bring up an agent diagram
• Invoke change-view by right clicking on the diagram
• Invoke change-view again

Results expected
• The relationship connectors between the component entities should vanish. All relations should be displayed as text within the components
• The relationship connectors the component entities should re-appear

10.9. Test Case 9 – Maintain object model

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.