Applying Broadcasting/Multicasting/Secured Communication to agentMom in Multi-Agent Systems

Test Plan

Version 1.1

This document is submitted in partial fulfillment of the requirements for the degree MSE.

Chairoj Mekprasertvit
CIS 895 – MSE Project
Kansas State University
Summer 2003

Table of Contents

Test Plan

1. Test Plan Identifier

TestPlan-agentMom-001

2. Introduction

This test plan is used to address the requires tests to show that the agentMom framework after the integration of broadcast, multicast and security features satisfies the requirements stated in the Software Requirements Specification version 1.0

2.1 Objectives

a.) To detail the activities required to prepare for and conduct the test
b.) To define the test cases needed to be performed
c.) To define the types of tests that will be used for each test cases
d.) To define the environment needed to perform the test

3. Test Items

The executable java classes to be tested are identified below:

a.) Conversation class
b.) MulticastConversation class
c.) BroadcastConversation class
d.) SecureUnicastConversation class
e.) SecureMulticastConversation class
f.) MulticastHandler class
g.) BroadcastHandler class
h.) SecureMulticastHandler class
i.) MessageHandler class
j.) SecureUnicastHandler class

4. Features to be tested

The following list describes the features that will be tested:

Specification Number	Description
T-001	Sending unicast message
T-002	Sending multicast message
T-003	Sending Broadcast Message
T-004	Sending secured unicast message
T-005	Sending secured multicast message
T-006	Subscribe to multiple multicast group
T-007	Receiving unicast message
T-008	Receiving multicast message
T-009	Receiving broadcast message
T-010	Receiving secured unicast message
T-011	Receiving secured multicast message
T-012	Encrypting message

T-013	Decrypting message

5. Features not to be tested

The test cases will not cover all possible size and value of sent message. Only possible size and value that are known to be required by project committee will be tested. Also, The test cases will not cover all possible combined features. Only classes that are related will be performed integration testing.

6. Approach

Unit testing – each executable java class identified in section 3 will be tested. One or more stub modules will be created to test functionality of each class. Junit 3.8 will be the tool to perform testing. Unit testing will be performed before alpha release.

Integration testing – Several related classes will be tested together to ensure sufficient execution and compliance with the requirements after integration. One or more stub modules will be created to test functionality of combined classes. Integration testing will be performed before beta release. In this test, two architectures, component-based and agent-based, are to be considered.

System testing – The whole system will be used for system testing to ensure all requirements is satisfied, and reliability will be included in the testing to measure successful rate of message delivery. Simple multi-agent systems will be created to perform system testing. System testing will be performed before final release.

7. Environmental needs

7.1 Hardware

The testing will be done on the CIS computer lab at Kansas State University. Furthermore, the testing will be done on the Sun Sparc machine and Intel-based machine available in the computer lab.

7.2 Software

j2sdk version 1.4 is used to compile and execute the program.

7.3 Operating Systems

1.) Windows XP professional
2.) Windows 2000
3.) Unix Solaris

8. Test Cases

Unit testing:

8.1 Sending and receiving unicast message

Input: Message Object
Test Item: Conversation class
Method: Create sender agent and receiver agent. Sender agent sends Message object to receiver agent through unicast conversation.

Valid:
Received Message object is the same as sending Message object.
Invalid:
Received Message object is not the same as sending Message object.

8.2 Sending and receiving multicast message

Input: Message Object
Test Item: MulticastConversation class
Method: Create sender agent and two receiver agents. All of agents subscribe to the same multicast address. Sender agent sends Message object to the receiver agents through multicast conversation.
Valid:
All receiver agents receive Message object
All received Message objects are the same as sending Message object.
Invalid:
One or more agents do not receive Message object
One or more received Message objects are not the same as sending Message object.

8.3 Sending and receiving Broadcast Message

Input: Message Object
Test Item: BroadcastConversation class
Method: Create sender agent and two receiver agents. All of agents are under the same local network. Sender agent sends Message object to the receiver agents through broadcast conversation.
Valid:
All receiver agents receive Message object
All received Message objects are the same as sending Message object.
Invalid:
One or more agents do not receive Message object
One or more received Message objects are not the same as sending Message object.

8.4 Sending and receiving secured unicast message

Input: Message Object
Test Item: SecureUnicastConversation class
Method: Create sender agent and receiver agent. Sender agent sends Message object to receiver agent through secured unicast conversation.
Valid:
Received Message object is the same as sending Message object after decrypting.
Message object is encrypted before sending.
Invalid:

Received Message object is not the same as sending Message object after decrypting.

Message object is not encrypted

8.5 Sending and receiving secured multicast message

Input: Message Object

Test Item: SecureMulticastConversation class

Method: Create sender agent and two receiver agents. All of agents subscribe to the same multicast address. Encryption and decryption key are pre-defined. Each agent has the same encryption and decryption key. Sender agent sends Message object to the receiver agents through secured multicast conversation.

Valid:

All receiver agents receive Message object

All received Message objects are the same as sending Message object.

Invalid:

One or more agents do not receive Message object

One or more received Message objects are not the same as sending Message object.

8.6 Encrypting and decrypting message

Input: Message Object

Test Item: SecurityManager class

Method: Create agent to read Message object. Input the Message object to SecurityManager class.

Valid:

Message object is unreadable after it is encrypted.

Message object is readable after it is decrypted.

Invalid:

Message object is readable after it is encrypted.

Message object is unreadable after it is decrypted.

Integration testing:

8.7 Subscribe to multiple multicast group plus agent-based architecture

Input: Message Object

Test Item: MulticastHandler and MulticastConversation class

Method: Create sender agent and two receiver agents. All of agents subscribe to three different multicast addresses. Sender agent sends Message object to the three multicast addresses through multicast conversation.

Valid:

All receiver agents receive all Message object.

All received Message objects are the same as Sending Message objects.

Invalid:

One or more agents do not receive one or more of Message objects.

One or more received Message objects are not the same as sending Message objects.

8.8 Subscribe to multiple multicast group with multicast security plus agent-based architecture

Input: Message Object

Test Item: SecureMulticastConversation, MulticastConversation and MulticastHandler class

Method: Create sender agent and two receiver agents. Encryption and decryption key are pre-defined. Each agent has the same encryption and decryption key. All of agents subscribe to two multicast addresses, one for multicast conversation and another one for secured multicast conversation. Sender agent sends Message object to the receiver agents through multicast conversation and secured multicast conversation.

Valid:

Message object is encrypted before sending.

Message object is decrypted after receiving.

All receiver agents receive all Message objects

All received Message objects are the same as Sending Message object.

Invalid:

Message object is not encrypted before sending.

Message object is not decrypted after receiving.

One or more agents do not receive Message object.

One or more received Message objects are not the same as sending Message object.

8.9 Subscribe to multiple multicast group plus component component-based architecture

Input: Message Object

Test Item: MulticastHandler and MulticastConversation class

Method: Create sender agent and two receiver agents. Each agent has two components. All of agents subscribe to three different multicast addresses. Sender agent sends Message object to the three multicast addresses through multicast conversation.

Valid:

All receiver agents receive all Message object.

All received Message objects are the same as Sending Message objects.

Invalid:

One or more agents do not receive one or more of Message objects.

One or more received Message objects are not the same as sending Message objects.

8.10 Subscribe to multiple multicast group with multicast security plus component-based architecture

Input: Message Object

Test Item: SecureMulticastConversation, MulticastConversation and MulticastHandler class

Method: Create sender agent and two receiver agents. Each agent has two components. Encryption and decryption key are pre-defined. Each agent has the same encryption and decryption key. All of agents subscribe to two multicast addresses, one for multicast conversation and another one for secured multicast conversation. Sender agent sends Message object to the receiver agents through multicast conversation and secured multicast conversation.

Valid:

Message object is encrypted before sending.

Message object is decrypted after receiving.

All receiver agents receive all Message objects

All received Message objects are the same as Sending Message object.

Invalid:

Message object is not encrypted before sending.

Message object is not decrypted after receiving.

One or more agents do not receive Message object.

One or more received Message objects are not the same as sending Message object.

System testing:

8.11 Test all features using agent-based architecture

Input: Message Object

Test Item: all items identified in section 3

Method: Create sender agent and two receiver agents. One agent performs encryption and decryption key distribution. Each agent requests the key from the specified agent. All of agents subscribe to two multicast address, one for multicast conversation and another one for secure multicast communication. Each agent requests the key from specified agent. Sender agent sends Message object to the receiver agents through unicast conversation, secured unicast conversation, multicast conversation, secured multicast conversation and broadcast conversation.

Valid:

All receiver agents receive all Message objects.

All received Message object is the same as sending Message object.

Invalid:

Some receiver agents do not receive all Message objects.

Some received Message object is not the same as sending Message object.

8.12 Test all features using component-based architecture

Input: Message Object

Test Item: all items identified in section 3

Method: Create sender agent and two receiver agents. Each agent has components. One agent performs encryption and decryption key distribution. Each agent requests the key from the specified agent. All of agents subscribe to two multicast address, one for multicast conversation and another one for secure multicast communication. Each agent requests the key from specified agent. Sender agent sends Message object to the receiver agents through unicast conversation, secured unicast

conversation, multicast conversation, secured multicast conversation and broadcast conversation.

Valid:

All receiver agents receive all Message objects.

All received Message object is the same as sending Message object.

Invalid:

Some receiver agents do not receive all Message objects.

Some received Message object is not the same as sending Message object.

Compatibility Testing:

8.13 Test backward compatibility of new agentMom and agentMom 1.2

Test Item: new agentMom

Method: Multi-agent systems that can be run on agentMom 1.2 should be able to run on new agentMom. The source of multi-agent systems will be supplied by Dr. DeLoach, and will test on new agentMom.

Valid:

Supplied systems must be able to run as same as under agentMom 1.2 without modifying the source code.

Invalid:

Supplied systems fails to run under the new agentMom.

9. Schedule

The testing will be performed during January 1, 2003 – January 18, 2003.