

Communication Model for Cooperative Robotics Simulator

Architecture Design

Version 1.0

1. Introduction

The purpose of this document is to provide architecture design including class diagram, class diagram description, sequence diagram and sequence diagram description of “Communication Model for Cooperative Robotics Simulator”. The architecture design is based on the Software Requirement Specification Version 1.1.

2. Class Diagram

There are eight classes in the system. The `CommunicationsSystem` class is served as a main interface, which interact with the environment. It provides function for robot and environment control panel.

The functions for robot are used to communicate with other robots, which are `registerRobot`, `sendMessage` and `getMessage`. Each simulated robot needs to register to the communication model to initiate the communication session and set what communication type is allowed, broadcast or point-to-point. When the environment takes requests from robot, it will pass them to the communication model by calling `sendMessage` in every time step. After communication model gets request to send message from the environment, it will process the message and add it to the receiver's priority queue. Same as `sendMessage`, the environment will call `getMessage` in every time step and pass the returned messages to the specified robot.

On the other hand, the functions for environment control panel are used to set robot's and system's parameter. The system parameters include system link status, system range, system delay and system delivery probability. The robot parameters are incoming and outgoing link status, maximum sending range, delay, delivery probability, broadcast and point-to-point ability.

The `RobotCommRecord` class is used to keep current robot parameter. Each `RobotCommRecord` has its own priority queue to keep messages, which will be retrieved when `getMessage` function is called. In other words, priority queue keeps a list of messages, which will be delivered to the corresponding robot

The `PriorityQueue` class keeps a set of messages ordered by `receivedTime`. After messages are delivered, they will be removed from the queue.

Since some parameters are defined as a value for each pair of robot, the `RobotParameter` class is created to keep the parameter for the owner robot and the other robots.

The `Message` class defines the sender, receiver, message content, sent time and received time. The sent time is the time step that message is sent by the environment to the communication model. The received time is the expect time step that robot will get the message.

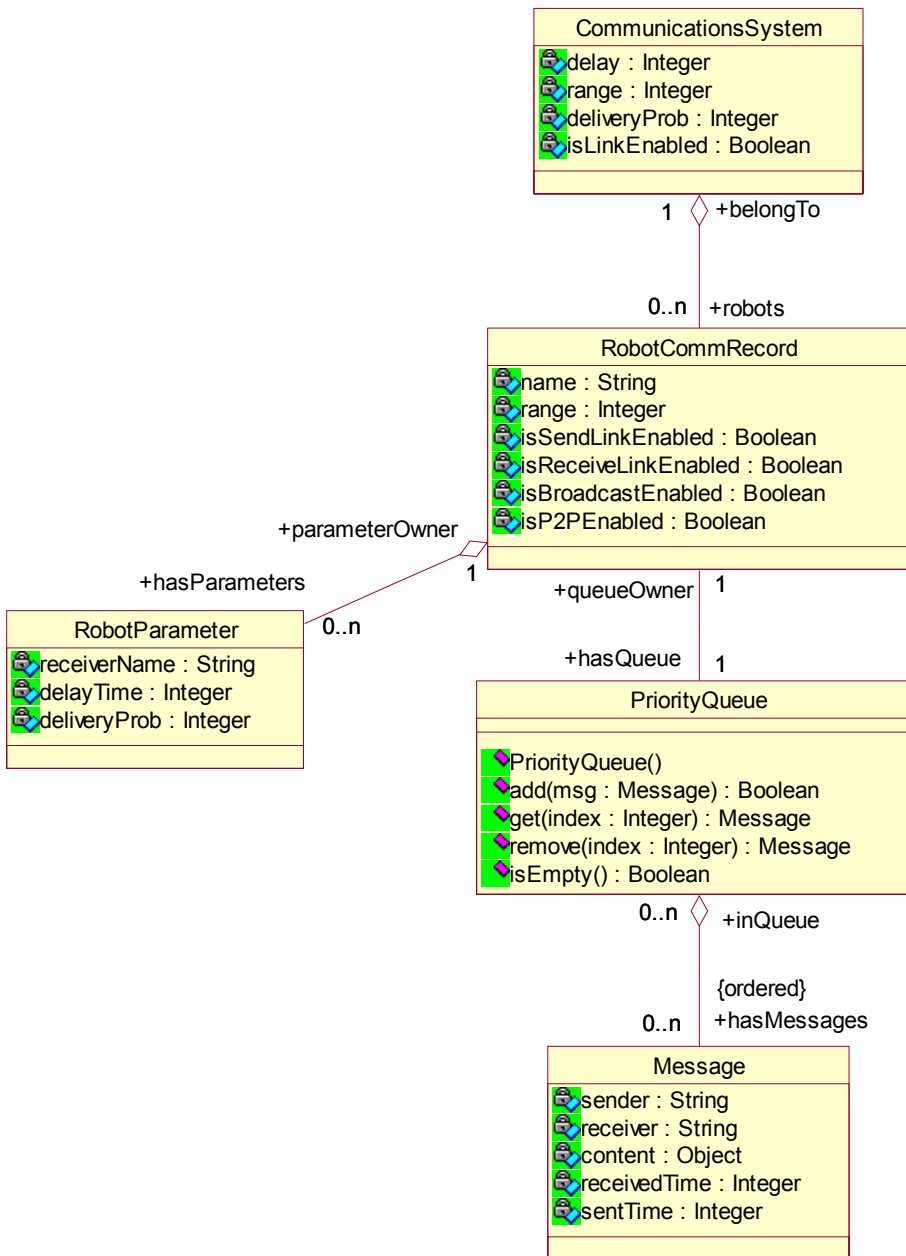


Figure 1. Class Diagram for Communication Model for Cooperative Robotic Simulator.

3. Sequence diagram

This section shows the sequence diagrams of the basic scenarios of Communication Model, which are register robot, send message and get message. The send message scenario includes two cases: sending broadcast and sending point-to-point

3.1 Register robot

After every robot registers to the environment, the environment will send the request to register robot with robot name and communication type to the CommunicationsSystem. If this robot name does not exist in the system, RobotCommRecord object will be created and set name as robot name. Then it will create their own priority queue and set communication ability: broadcast, point-to-point or both depends on the communication type.

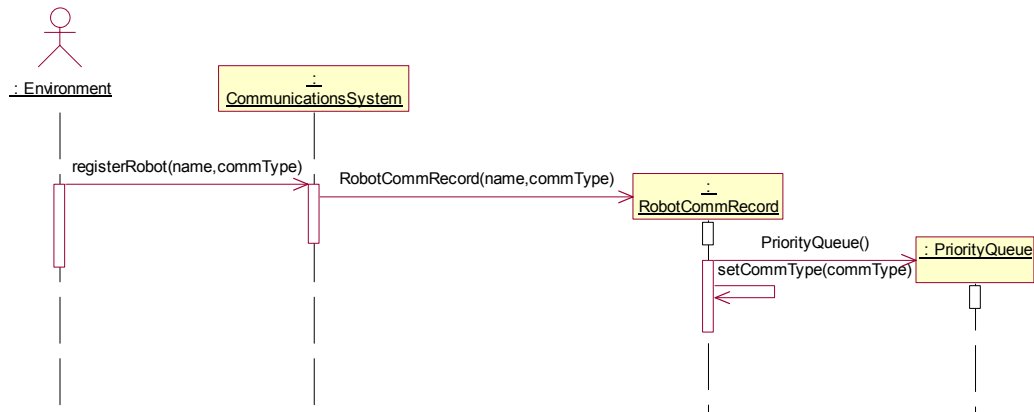


Figure 2. Register Robot Sequence Diagram

3.2 Send message (broadcast)

Environment sends a request to send message with message and time step as parameters. If the system link is enabled, then it will check if this sender has the ability to send broadcast message. If the sender has ability to send broadcast message, it will check each robot in the system if they qualify to get this broadcast message. If they qualified, the message will be added to their priority queue.

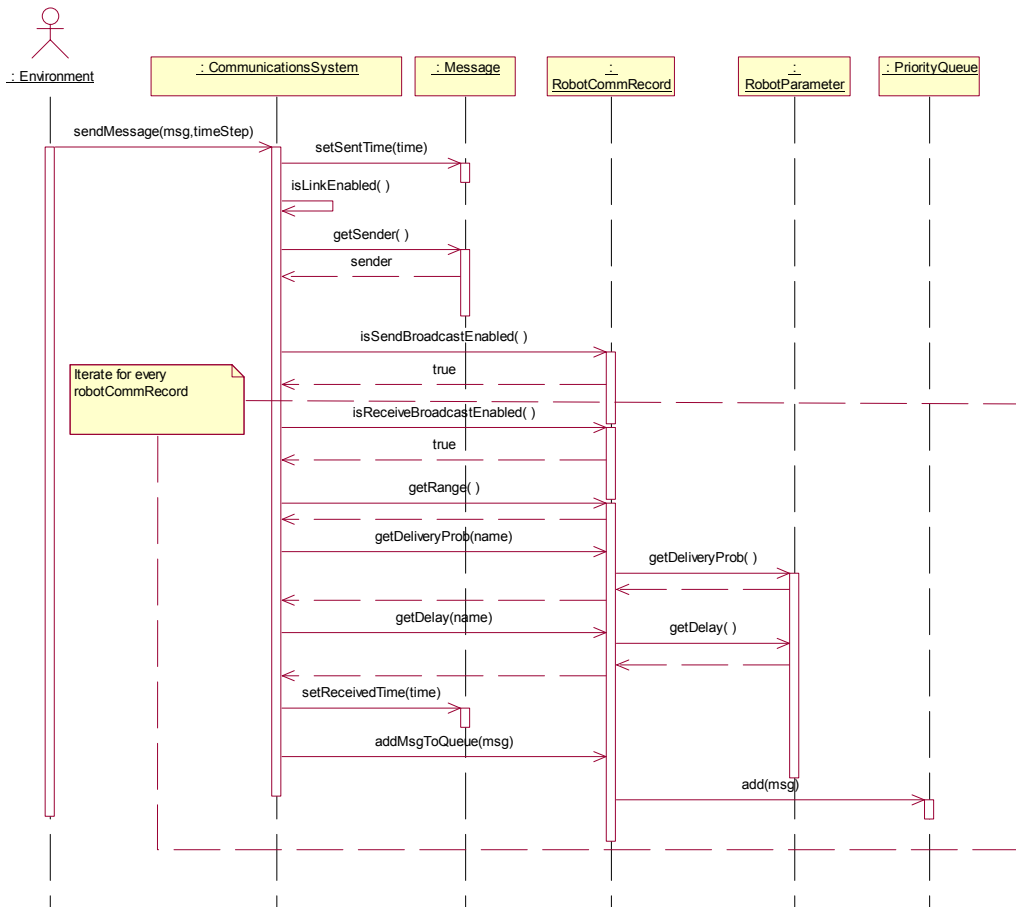


Figure 3. Send Message (Broadcast Message) Sequence Diagram

3.3 Send message (point-to-point)

Environment sends a request to send message with message and time step as parameters. If the system link is enabled, then it will check if this sender has the ability to send point-to-point message. If the sender has ability to send point-to-point message, it will check if the receiver qualifies to get this message. If it qualifies, the message will be added to the priority queue.

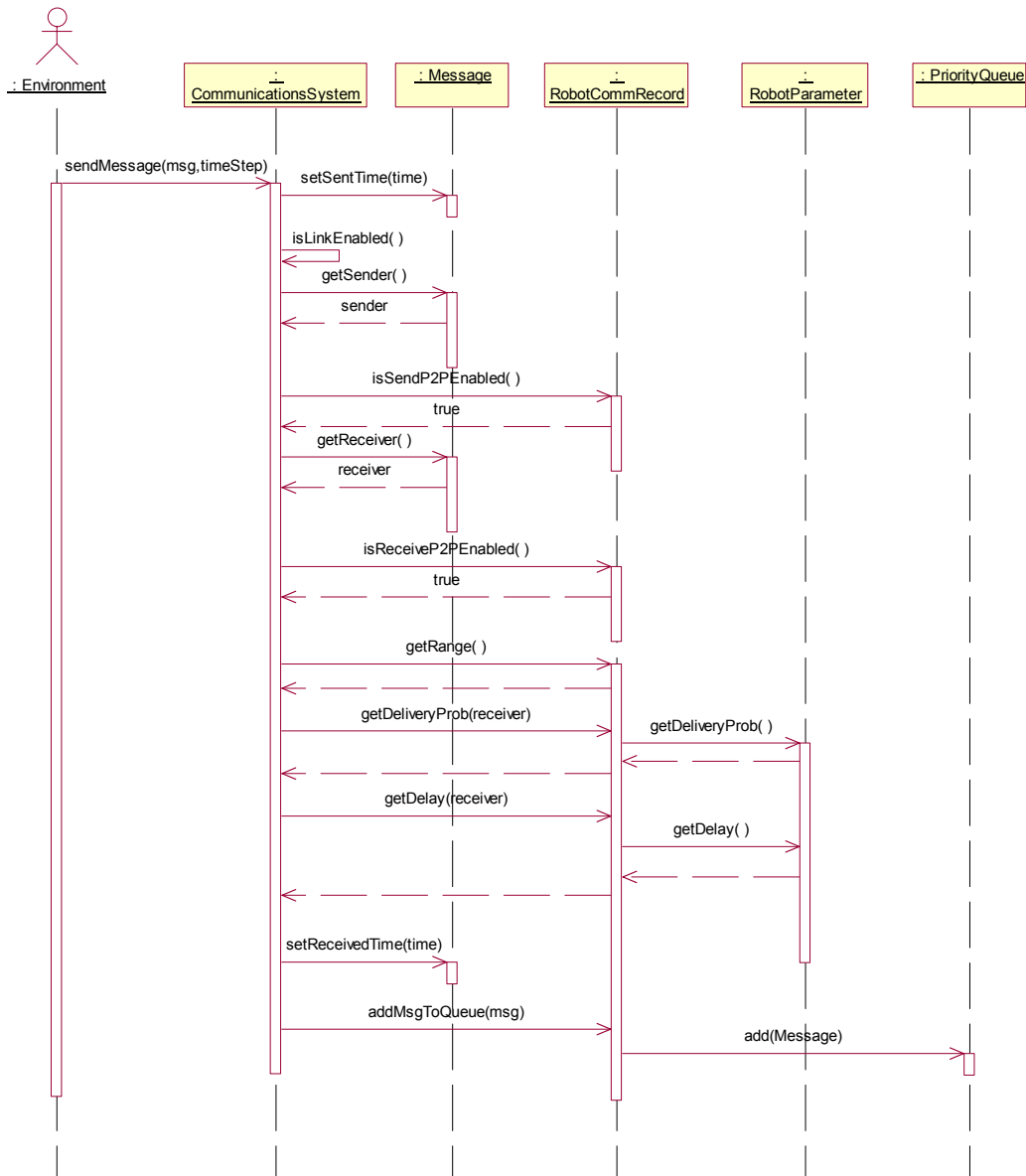


Figure 4. Send Message (Point-to-point Message) Sequence Diagram

3.4 Get message

Environment sends request to retrieve message for each robot in every time step. The system will retrieve the messages which has received time equals to time step from the robot's priority queue.

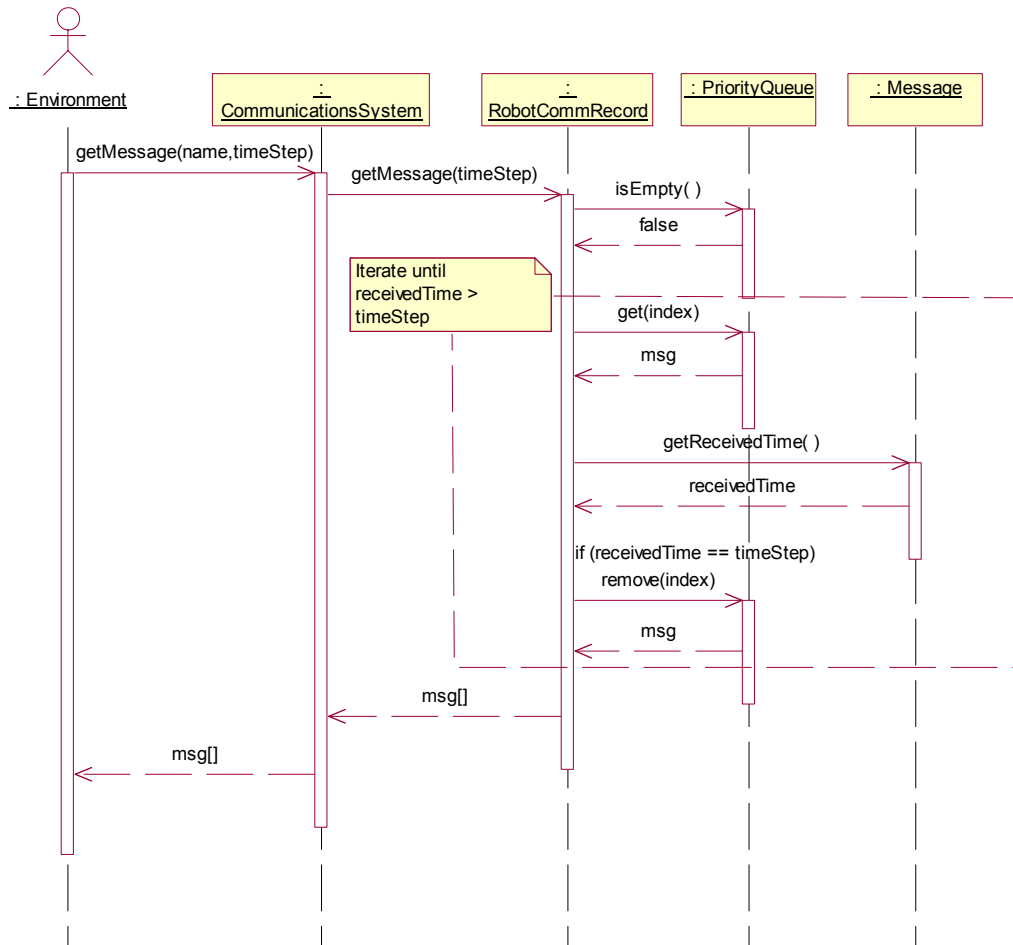


Figure 5. Get Message Sequence Diagram

4. Class Description

This section provides class description in detail.

4.1 CommunicationsSystem

This class is the main interface of the system, which provides a set of methods used by Environment. The environment uses the interface for sending/receiving messages and passes them to the targeted robot. The environment also uses a set of methods to set and get parameter values and pass to environment control panel. The parameters are system parameters, and robot parameters. This class also used to keep system parameter including range, link, delivery probability and delay.

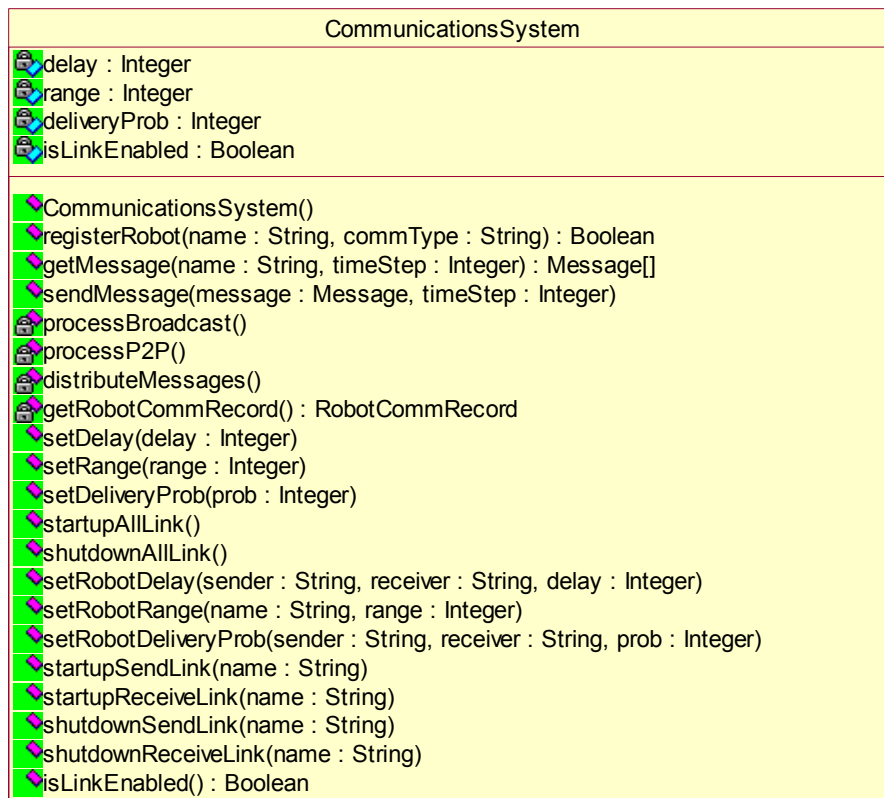


Figure 6. CommunicationsSystem class with attributes and operations.

4.2 RobotCommRecord

This class is used to keep robot parameters, which are range, active link, delivery probability, delay, and broadcast or point-to-point ability. It provides methods to set these parameters. In addition, it validates each incoming message, and distribute message to the specified robot.

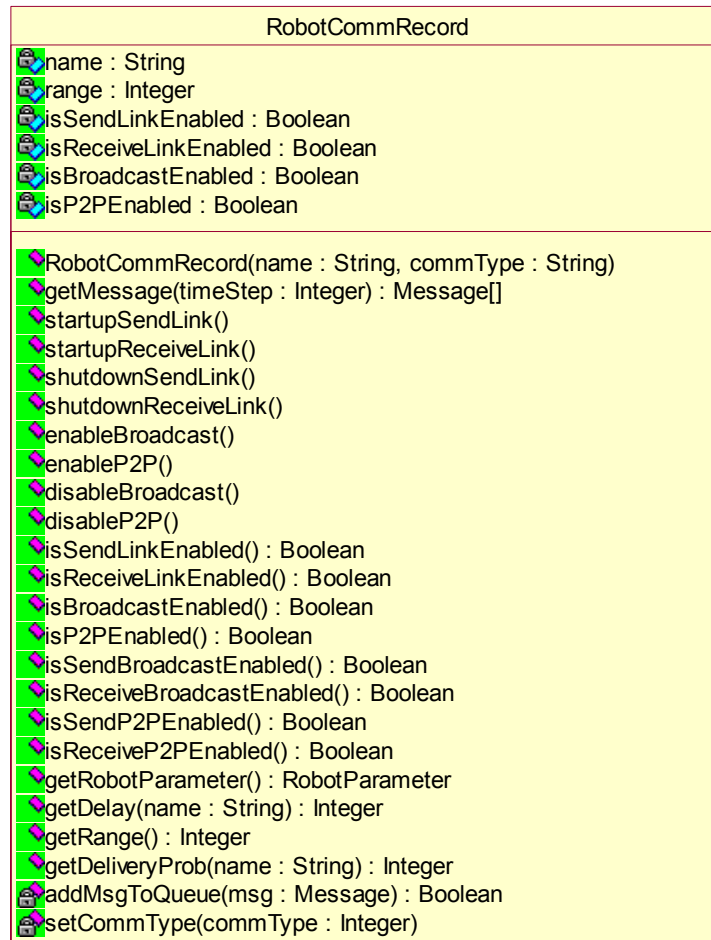


Figure 7. RobotCommRecord class with attributes and operations.

4.3 RobotParameter

It keeps the parameters for each pair of robot (the owner and the other robots).

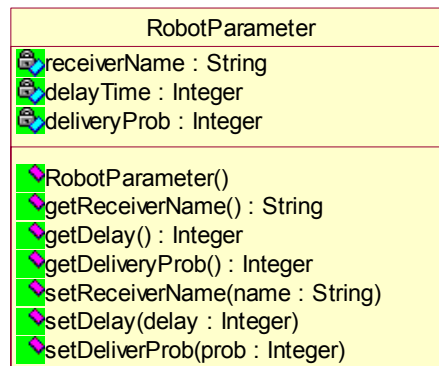


Figure 8. RobotParameter class with attributes and operations.

4.4 PriorityQueue

This class is the priority queue, which will keep outgoing messages (in terms of this system, but it is the incoming message for robot) ordered by received time. It also provides method to add and remove from queue.

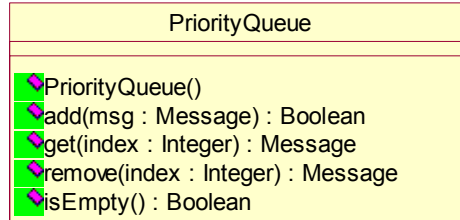


Figure 9. PriorityQueue class with attributes and operations.

4.5 Message

This class defines the field use to communicate between robots including sender, receiver, sent time, received time, and message content.

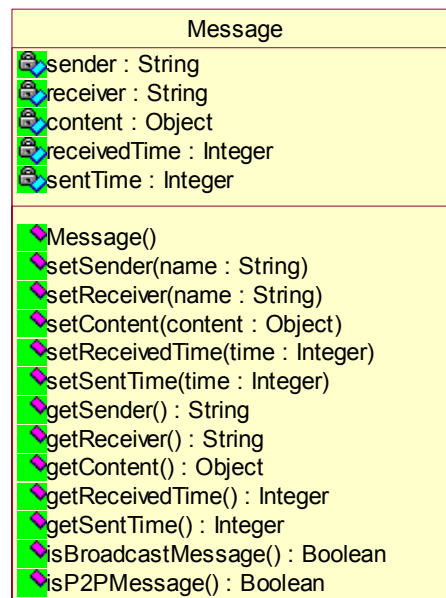


Figure 10. Message class with attributes and operations.

5. CommunicationsSystem State Diagram

The State Diagram shows the possible ways in which the objects respond to events, which occur in the system. There are five states in the CommunicationsSystem State Diagram, wait, registering, sending, sending broadcast, sending point-to-point, adding to queue and getting message.

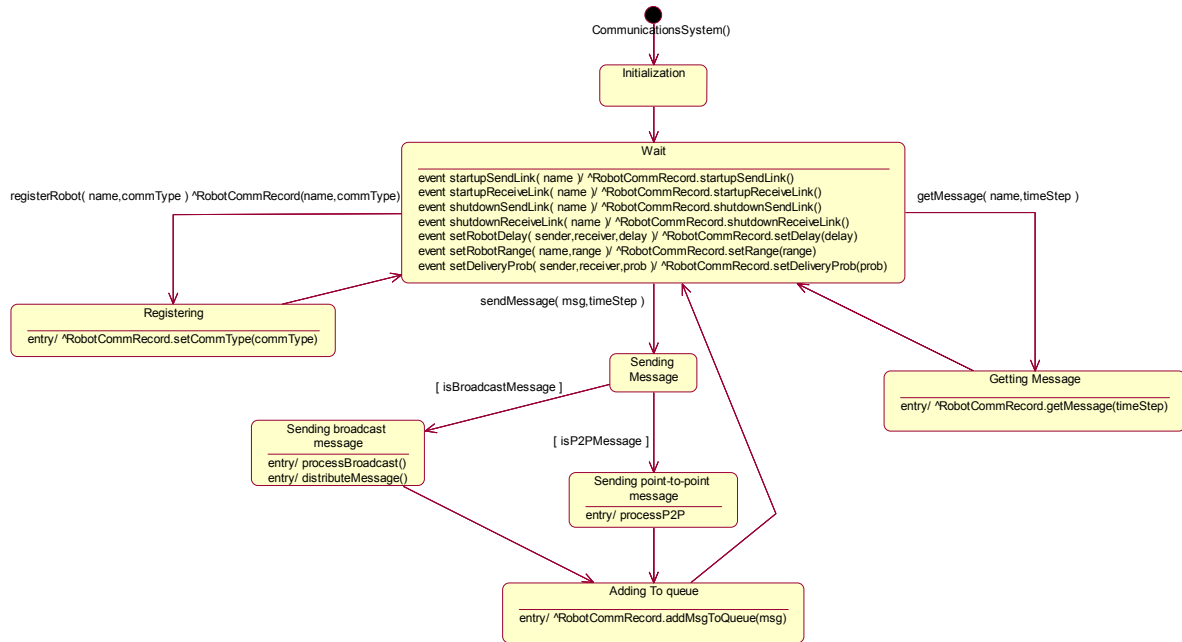


Figure 11. CommunicationsSystem State Chart Diagram.

6. Reference

Lee, Richard C., Tevfendhart, William M., *Practical Object-Oriented Development With UML and Java*, Prentice Hall, 2002