Communication Model for Cooperative Robotics Simulator

# Formal Technical Inspection Checklist

Version 1.0

---

## 1. Introduction

The purpose of this document is to provide formal technical inspection checklist of the design architecture for "Communication Model for Cooperative Robotic Simulator". Two independent MSE students will inspect the design architecture and provide a report on the result of their inspection. The formal inspection is the review process to ensure the quality of the software design, which will be useful toward the development process.

## 2. Item to be inspected

The architecture design of "Communication Model for Cooperative Robotic Simulator" will be inspected including use-case diagram, class diagram and sequence diagram. Some reference documents will be provided to give some background of the system. The reference documents are Software Requirement Specification version 1.0 and Project Overview version 1.0.

The following are the architecture design will be inspected.
- Use case diagram
- Class diagram
- Sequence diagram

## 3. Formal technical inspector
- Kevin Sung
- Esteban Guillen

## 4. Formal technical inspection checklist

| Inspection list | Pass/Fail/Partial | Comment |
|---|---|---|
| 1. The symbols using in use case diagram conform to UML diagram. | Pass | It looks good |
| 2. The symbols using in class case diagram conform to UML diagram. | Pass | It looks good |
| 3. The symbols using in sequence diagram conform to UML diagram. | Pass | It looks good |
| 4. Use case diagram and descriptions are clear and well organized. | Pass | Yes it was clear to me |
| 5. Class diagram and descriptions are clear and | Pass | I had a few |

| well organized. | | questions. I didn't see a constructor for the RobotParameter class and was wondering how it would get created. Also in the RobotParamater class there are four "get" methods but there are no return types shown. |
| --- | --- | --- |
| 6. Each message passing in sequence diagram is the method in class diagram. | Pass | Yes it matches up |
| 7. Each message passing in sequence diagram must be defined as public method. | Pass | Yes |
| 8. Class names are well defined and indicate their meaning | Pass | Yes |
| 9. The architecture design covers the entire requirement defined in Software Requirement Specification. | Pass | Yes nice work. I just had a question about section 4.3 in your Architecture Design doc. The sequence diagram looks good, but the description above talks about sending broadcast messages while the diagram is showing point-to-point. |