
Software Requirements Specification

for

Multi-Agent Communication

Version 1.0

Prepared by Melike ARSLAN

Bilkent University

19.06.2019

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction.....	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	1
2. Overall Description	1
2.1 Product Perspective	1
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	3
2.5 Design and Implementation Constraints	3
2.6 User Documentation	3
2.7 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
4. System Features.....	4
4.1 One-Way Communication	4
4.2 Multi-Way Communication	4
4.3 Broadcast Communication	5
5. Other Nonfunctional Requirements	6
5.1 Performance Requirements	6
5.2 Safety Requirements	6
5.3 Security Requirements	6
5.4 Software Quality Attributes	6
5.5 Business Rules	7
6. Other Requirements	7
Appendix A: Glossary.....	7
Appendix B: Analysis Models.....	7
Appendix C: To Be Determined List	7

Revision History

Name	Date	Reason For Changes	Version
Melike ARSLAN	19.06.19	Initial release	0.1

1. Introduction

1.1 Purpose

This Software Requirements Specification describes the functional and nonfunctional requirements for Multi-Agent Communication 1.0. Multi-Agent Communication 1.0 will be a multi-agent communication module to make soft communication between agents and coordinators. Unlike other multi-agent systems (MAS), Multi-Agent Communication 1.0 will focus specifically on communication.

1.2 Document Conventions

No document conventions are being used at this time.

1.3 Intended Audience and Reading Suggestions

The intended audience of this system is the development team who will integrate this system with their own systems creating a bigger project on multi-agent co-operation. However, other developers, testers or users who focus on multi-agent communication may use this system as well.

The rest of this SRS consists of the overview of the project and general requirements of the project. The general requirements are explained in more detail in each of their sections, starting with external interface requirements.

1.4 Product Scope

Multi-Agent Communication 1.0 is a communication module for specifically multi-agent systems. It allows agents to communicate in different types of ways. These types are detailed in Section 4 of this document.

1.5 References

[IEEE] SRS Standardization 830, 1998 edition.

2. Overall Description

2.1 Product Perspective

Multi-Agent Communication 1.0 is a new system intended to be a part of a larger system that involves task planning and scheduling, path planning and drone path planning. Each component uses AI techniques.

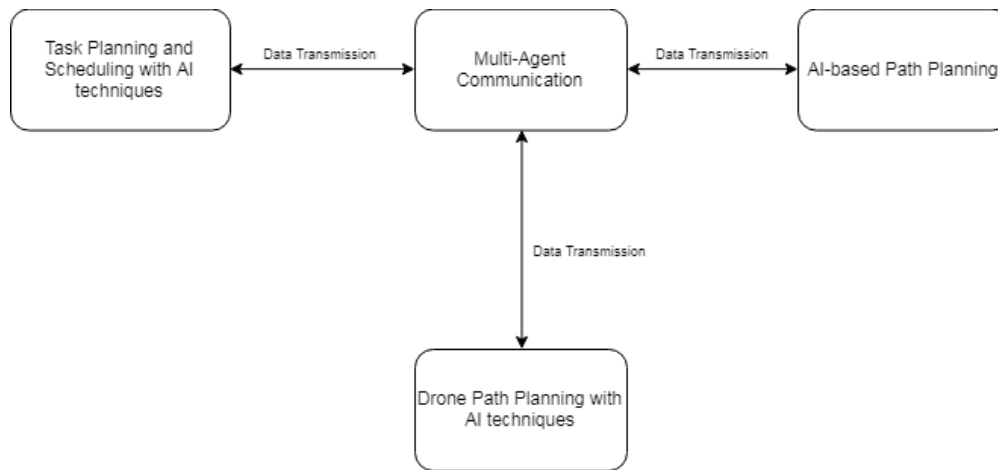


Figure 1: Context diagram for the general system

The context diagram in Figure 1 illustrates the relationships between each component of the general system. The Multi-Agent Communication system is the critical component because without it, there is no way for the other systems to communicate with each other. Task Planning and Scheduling with AI techniques System provides the initial data given to the Multi-Agent Communication System. The Multi-Agent Communication System transmits this data according to the content to either one of the path planning systems and the Multi-Agent Communication receives feedback from these systems and returns it back to the task planning system.

2.2 Product Functions

The features of the system are

- One-Way Communication
- Multi-Way Communication
- Broadcast Communication

These features will be explained in detail in Section 4.

2.3 User Classes and Characteristics

Agent	An Agent is tasked with certain duties. Each agent is expected to complete these duties and report back to the coordinator.
Coordinator	A coordinator is also an agent inheriting the attributes of an agent. Aside those attributes, it can broadcast data which ordinary agents are not allowed to do.

2.4 Operating Environment

System shall run on computer platforms. Operating systems, the system can operate on is

- Ubuntu 18.04

2.5 Design and Implementation Constraints

The system shall be implemented in Python 3.7. Frameworks that are used in multi-agent systems can be used depending on the licensing they use. Details of the framework that will be used are to be determined and explained in the design phase of the project. There are no time or memory constraints as of now.

2.6 User Documentation

No user documentation is available at this time.

2.7 Assumptions and Dependencies

It is assumed that the other systems that may use Multi-Agent Communication 1.0 are also implemented in Python 3.7. It is also assumed that the user has Ubuntu 18.04. It is assumed that the frameworks this system will use have no major bugs.

3. External Interface Requirements

3.1 User Interfaces

UI-1: To show the communication between the agents the user shall be provided with a screen layout that resembles a messaging chat box.

UI-2: The system shall provide the information being transmitted on the screen.

UI-3: The system shall prompt the user to send the message again if the transmission was not successful.

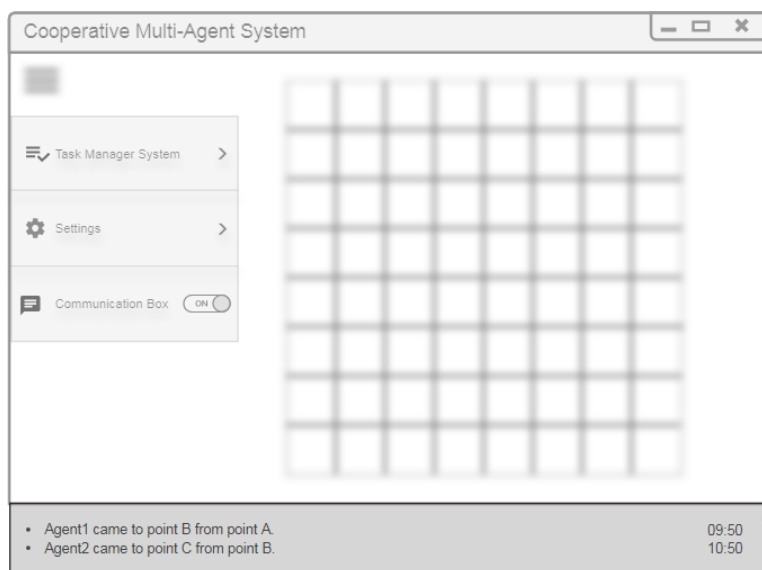


Figure 2: Mockup for the general user interface with the message box below the layout

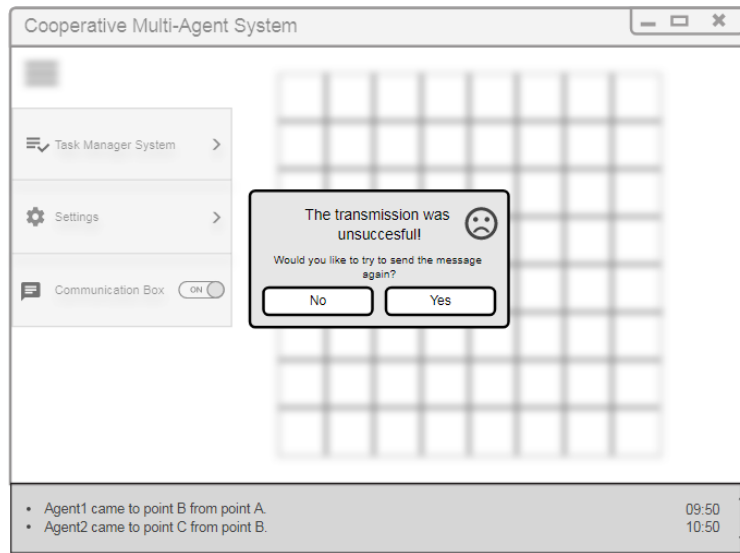


Figure 3: Mockup for the general user interface with unsuccessful message prompt

3.2 Hardware Interfaces

- HI-1: Keyboard
- HI-2: Screen
- HI-3: Mouse

3.3 Software Interfaces

- SI-1: The operating system shall be Ubuntu 18.04.
- SI-2: The Integrated Development Environment (IDE) shall be PyCharm.

More software interface requirements are to be determined such as frameworks, libraries or tools in the development phase of the project.

3.4 Communications Interfaces

- CI-1: The system shall follow a publisher/subscriber method since it will be asynchronous.
- CI-2: The system shall acquire coordination information from the task manager system.
- CI-3: The system shall direct the coordination information to the drone path planning system.
- CI-4: The system shall acquire the information about the task needed on that coordination from the drone path planning system.
- CI-5: The system shall direct the task information to the task manager system.
- CI-6: If there is no route or the drone has low battery, the system shall acquire the information that the task of the drone is not completed.
- CI-7: The system shall direct the state of the success of the task to the task manager system.
- CI-8: The system shall acquire the information of the coordinates, the type of task and the number and type of the agent/agents needed for the specified task from the task manager system.

- CI-9: The system shall direct the information of the coordinates, the type of task and the number and type of the agent/agents needed for the specified task to the multi-agent path planning system.
- CI-10: The system shall acquire the information of the task completion success from the multi-agent path planning system.
- CI-11: The system shall direct the state of the success of the task to the task manager system.
- CI-12: If the location is unknown, the location is wrong which means there is an obstacle on the location, the number of agents exceeded, the number of agents is insufficient, there is no path and there is collision, the system shall inform the task manager system of these exceptions.

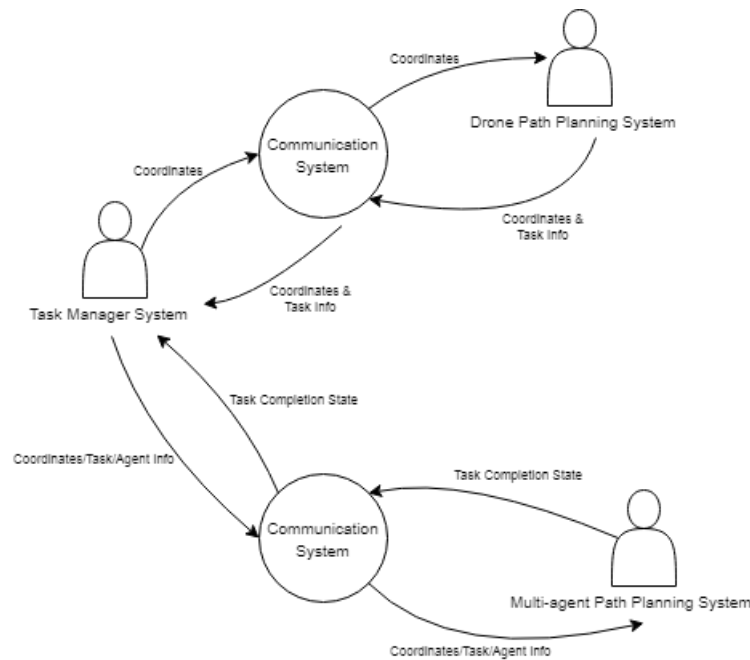


Figure 4: A simple diagram to summarize the communication requirements

4. System Features

4.1 One-Way Communication

4.1.1 Description and Priority

Agents shall have the ability to send and receive one-way messages. Each message shall particularly be received by a specified address and shall be sent to a specified address. The messages shall be ordered by the receiver agent according to the time the message was sent by the sender agent.

Priority: High

4.1.2 Stimulus/Response Sequences

Stimulus: Agent sends message to another agent.
 Response: System retrieves the message and sends it to the address of the receiver.

Stimulus: Agent receives message.
 Response: System retrieves the information of the state of the successfulness of the transmission.

4.1.3 Functional Requirements

REQ-1	Send Message	The system shall retrieve the message and the specified address of the receiver from the agent and send it to the receiver according to the address indicated.
REQ-2	Send Message Unsuccessful	The system shall check if the message is delivered to the receiver. If the delivery is unsuccessful, the system shall prompt the sender to send the message again.
REQ-3	Receive Message	The system shall retrieve the information of the state of the successfulness of the transmission.

4.2 Multi-Way Communication

4.2.1 Description and Priority

Agents shall have the ability to send and receive multi-way messages. Each agent shall have the ability to request to join and leave a multi-way group. In each group the message shall only be sent to the group members. Each agent shall have the ability to be a member of many groups. Each agent shall have the ability to create a multi-way group. The initial creator of the group shall have the ability to set a time period for the group to stay active.

Priority: Low

4.2.2 Stimulus/Response Sequences

Stimulus: Agent creates group.
 Response: System retrieves the address data of the members from the creator agent and stores the data.

Stimulus: Agent sets time for a group.
 Response: System keeps the group active until the given time runs out.

Stimulus: Agent sends multi-way message.
 Response: System directs the message to every member in the multi-way group.

Stimulus: Agent requests to join a group.
 Response: System alerts the creator agent and asks whether to let the agent in the group or not.

Stimulus: Agent leaves a group.

Response: System removes the agent from the group.

4.2.3 Functional Requirements

REQ-1	Create Group	The system shall create a group with the members specified by the creator agent.
REQ-2	Send Message	The system shall allow every member of the group to send a multi-way message.
REQ-3	Set Time Period	The system shall allow the creator of the group to set a time period for the group in which the system shall delete the group when the time runs out.
REQ-4	Join Group	The system shall alert the creator that an agent wants to join the group.
REQ-5	Leave Group	The system shall allow an agent to leave a multi-way group. The system shall remove the agent as a member

4.3 Broadcast Communication

4.3.1 Description and Priority

Coordinator agent shall have the ability to send broadcast messages. Other agents shall have the ability to receive broadcast messages. Broadcast messages shall be sent to all the available agents in the system.

Priority: High

4.3.2 Stimulus/Response Sequences

Stimulus: Coordinator sends broadcast message.

Response: System retrieves the message and forwards it to all the other agents in the system.

4.3.3 Functional Requirements

REQ-1	Broadcast Message	The system shall allow the coordinator to broadcast a message to all the available agents.
-------	-------------------	--

5. Other Nonfunctional Requirements

5.1 Performance Requirements

PR-1: The system shall be responsive.

PR-2: The latency of the system shall be no more than xx (TBD).

PR-3: Memory requirements are to be determined in the design phase of the project.

5.2 Safety Requirements

No safety requirements have been identified.

5.3 Security Requirements

No security requirements have been identified.

5.4 Software Quality Attributes

Interoperability: The system shall be interoperable since it focuses on multi-agent communication.

Flexibility: The system shall be flexible so that it can interact with other systems.

Maintainability: The system shall be maintainable for later versions of the system.

Portability: The system shall be portable for different platforms such as Ubuntu operating system.

Robustness: The system shall be robust as long as the input is correct.

Reusability: The system shall be reusable because many different systems may need a communication system.

Testability: The system shall be testable with virtual devices in later phases of the project.

5.5 Business Rules

No business rules have been identified.

6. Other Requirements

No other requirements have been identified.

Appendix A: Glossary

Term	Definition
SRS	Software Requirements Specifications
MAS	Multi-agent System

IEEE	Institute of Electrical and Electronics Engineers
AI	Artificial Intelligence
IDE	Integrated Development Environment

Appendix B: Analysis Models

No pertinent analysis models included.

Appendix C: To Be Determined List

- Design and Implementation Constraints
- Software Interfaces
- Performance Requirements