Multi-agent Communication

Software Design Document

Version 1.0

Melike ARSLAN
24.06.19

1. Overview

This document contains the detailed software design descriptions for Multi-agent Communication (MAC). This document is prepared according to the “IEEE Standard for Information Technology - Systems Design - Software Design Descriptions - IEEE 1016 - 2009”.

This document provides the details of how MAC should be built which are indicated by using UML diagrams and illustrations.

1.1. Scope

MAC is a communication system designed to be used between agents. MAC acts as a layer for communication between the agents. It provides different communication types for the user to choose upon for the agents.

1.2. Purpose

This document visualizes the design and architecture of MAC. The main purpose is to represent the design of MAC with UML diagrams.

1.3. Intended Audience

The intended audience consists of developers who might use this system with a purpose of their own.

1.4. References

IEEE. IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2. Definitions

Term	Definition
MAC	Multi-agent Communication
IEEE	Institute of Electrical and Electronics Engineers
UML	Unified Modeling Language
SDD	Software Design Descriptions

3. Design Viewpoints

3.1. Introduction

In this section of the document, the viewpoints of MAC are explained using UML diagrams. The viewpoints consist of:

- Context Viewpoint
- Composition Viewpoint
- Logical Viewpoint
- Dependency Viewpoint
- Information Viewpoint
- Patterns use Viewpoint
- Interface Viewpoint
- Structure Viewpoint
- Interaction Viewpoint
- State Dynamics Viewpoint
- Algorithm Viewpoint
- Resource Viewpoint

3.2. Context Viewpoint

3.2.1. Design Concerns

There are two entities called publisher and subscriber. These entities are described as a method on how agents can communicate with each other.

3.2.2. Design Elements

The two entities publisher and subscribers are the actors in this system. The use-case diagram below illustrates this.

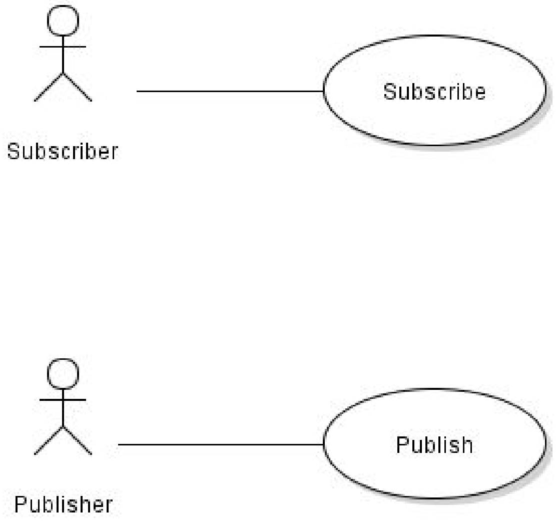


Figure 1: Use case diagram for the context viewpoint

3.2.2.1. Subscribe

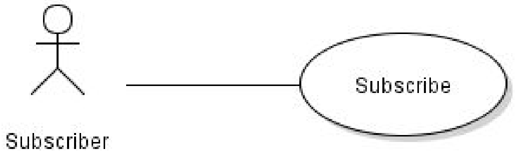


Figure 2: Use case diagram-Subscribe

Use Case Number	1
Use Case	Subscribe
Actor	Subscriber
Primary Scenario	Subscriber can subscribe to a publisher so whenever the publisher publishes, the subscriber will be informed of the data.

3.2.2.2. Publish

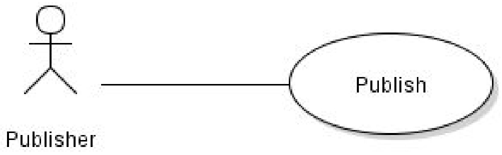


Figure 3: Use case diagram-Publish

Use Case Number	2
Use Case	Publish
Actor	Publisher
Primary Scenario	Publisher can publish data to all the subscribers in the subscriptions.

3.3. Composition Viewpoint

There is no need for composition viewpoint for this system.

3.4. Logical Viewpoint

3.4.1. Design Concerns

Logical viewpoint identifies the classes and the relations between the classes in MAC.

3.4.2. Design Elements

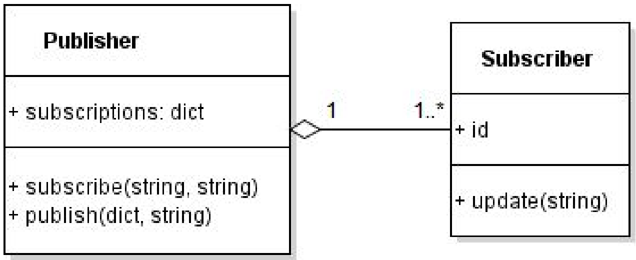


Figure 4: Class diagram

The class diagram indicates that there is “Publisher” class with an attribute called “subscriptions”. The “subscriptions” is a dictionary attribute which consists of all the subscribers to a specified publisher.
“Publisher” has two functions called “subscribe” and “publish”. The “subscribe” function takes two strings as parameters which indicates the name of a subscriber and the topic which the subscriber will be subscribed to. The “publish” function takes a dictionary and a string as parameters which indicates the data to be published and the topic on which it will be published as.
“Subscriber” class has an attribute called “id” which the type has not been identified yet. The “id” shall be unique. “Subscriber” has a function called “update” which simply acts as a print function to show that a publish has reached a subscriber.

3.5. Dependency Viewpoint

There is no need for dependency viewpoint for this system.

3.6. Information Viewpoint

There is no need for dependency viewpoint for this system.

3.7. Patterns use Viewpoint

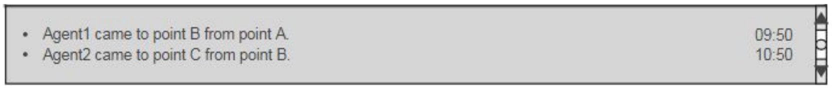
Design pattern used in this system is publish/subscribe pattern.

3.8. Interface Viewpoint

3.8.1. Design Concerns

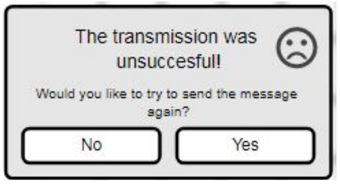
The interface viewpoint provides information on how the system will be seen as an interface.

3.8.2. Design Elements



- Agent1 came to point B from point A. 09:50
- Agent2 came to point C from point B. 10:50

Figure 5: Communication box mockup



The transmission was unsuccesful!

Would you like to try to send the message again?

Figure 6: Unsuccessful message mockup

3.9. Structure Viewpoint

There is no need for structure viewpoint for this system.

3.10. Interaction Viewpoint

3.10.1. Design Concerns

The interaction viewpoint provides information on the interactions between objects. Each interaction are represented scenarios. The scenarios are illustrated using sequence diagrams.

3.10.2. Design Elements

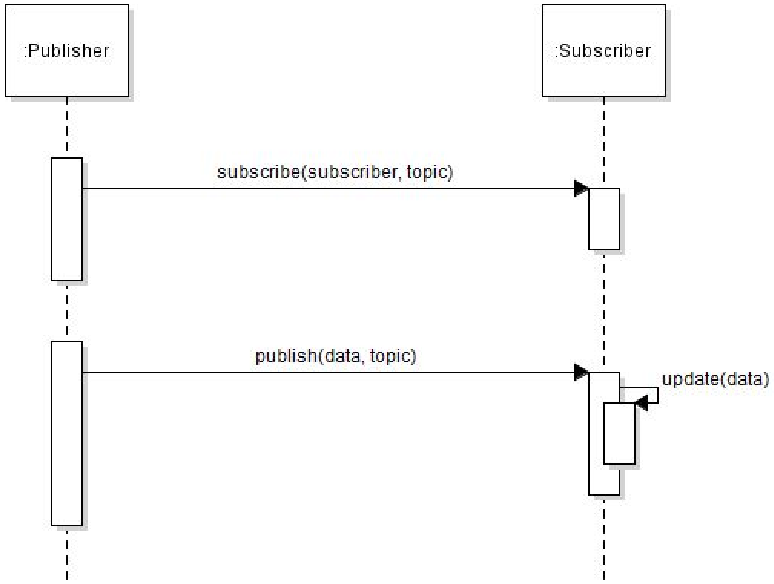


Figure 7: Sequence diagram

3.11. State Dynamics Viewpoint

3.11.1. Design Concerns

The state dynamics viewpoint provides information on the dynamic state transformations. The transformations are illustrated using state diagrams.

3.11.2. Design Elements

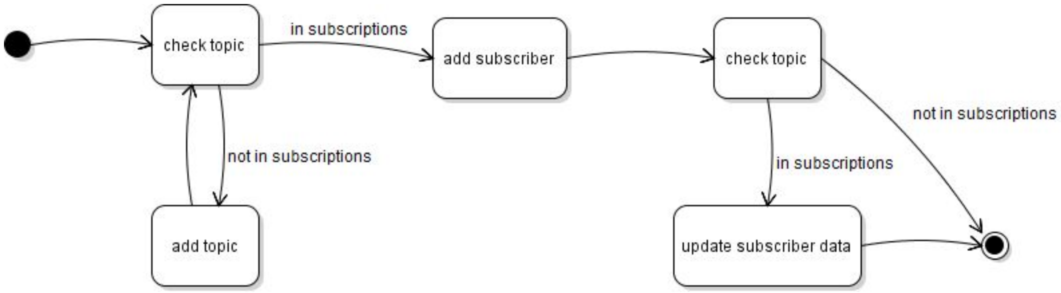


Figure 8: State diagram

3.12. Algorithm Viewpoint

There is no need for algorithm viewpoint for this system.

3.13. Resource Viewpoint

There is no need for resource viewpoint for this system.