Real Time Location System

Conceptualization Document

[Revision history]

Revision date	Version #	Description	Author
	0.01	First Documentation	
	0.02	Use case 추가 및 수정	

= Contents =

1. Business purpose

1.1. Project background

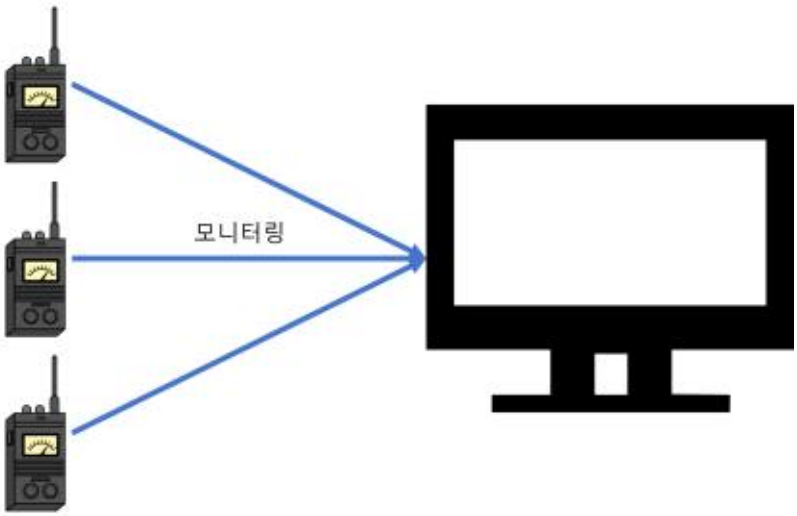
(그림 1) 호스를 옮기는 소방관

(그림 2) 도시화재를 진화하는 소방관

소방관들은 화재 등 각종 재난 현장으로 출동할 때 위험 상황 대비 및 통신을 위해 무전기를 들고 간다. 하지만 소방관은 재난현장에서 숨지거나 다치는 사례가 끊이지 않고 있다. 그중 구조작업 중에 실종되었다가 순직하는 경우도 많다.

이러한 안타까운 사례들을 방지할 수 있는 방법이 없을까 하던 중 “무전기에 GPS와 위험감지장치를 부착하여 중앙 센터에서 무전기들의 위치 및 상태를 실시간으로 모니터링하며 관리하면 어떨까?”라는 생각을 하였다.

그렇게 생각난 것이 바로 "Real Time Location System"이다.



(그림 3) Real Time Location System의 간단한 구조

이 시스템은 무전기들의 위치와 상태를 옵저버에서 실시간으로 확인 가능하며 실시간으로 무전기들의 위치와 상태를 관리함으로써 각종 재난 지역에서 위험 상황에 대해 대비할 수 있다.

하지만 실제로 무전기를 만들 수 없으므로 이를 프로그램으로 만들어 클라이언트(무

전기)는 키보드로 위치를 움직이며 그 위치를 실시간으로 모니터링 한다. 또한 무전기로 음성 통신하는 것은 채팅으로 대체하였다.

그렇게 우리는, 다음과 같은 Real Time Location System을 개발하여 이전 무전기보다 개선된 위험 상황에 대비할 수 있는 체계를 구축하고자한다.

- 클라이언트는 프로그램을 실행하여 시스템에 접속한다. 클라이언트 프로그램에는 가상의 재난 지역이 출력된다.
- 클라이언트는 키보드로 움직이며 위험한 장소에 들어가면 위험 감지 장치가 이를 감지하여 알린다.
- 옵저버에서는 클라이언트들의 실시간 위치와 상태를 확인하고 이동 경로를 확인함으로써 위험 상황에 보다 안전하게 대비할 수 있다.

이 시스템은, 각종 재난 지역에 구조 활동 등을 위해 출동하는 기관들을 대상으로, Real Time Location System을 개발하여 구조 활동을 하다가 순직하는 안타까운 상황들을 방지함과 동시에, 구조 활동하는 사람들의 위험상황에 보다 발 빠르게 대처할 수 있을 것으로 기대된다.

1.2. Target Market

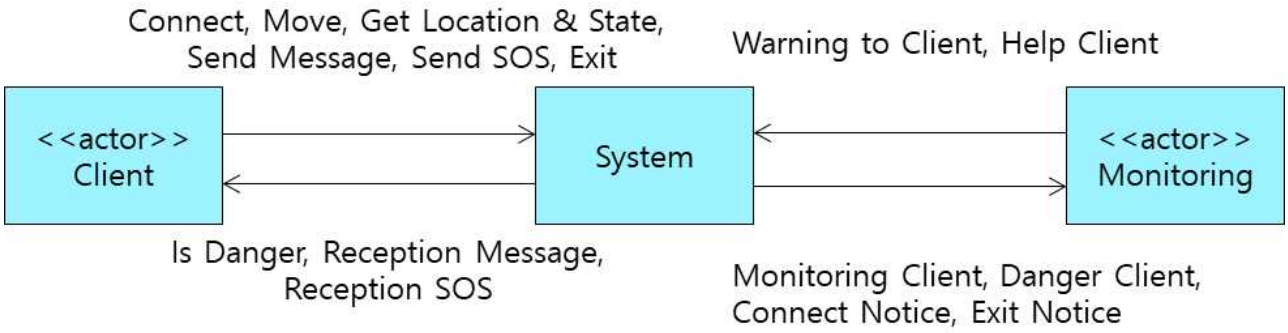
여러 위험한 지역에서 구조 활동을 하는 단체 또는 기관

1.3. Goals

클라이언트의 위치 및 상태를 실시간으로 모니터링 할 수 있는 프로그램 개발

클라이언트들의 위치 및 상태를 실시간으로 모니터링 할 수 있는 응용프로그램으로서, 클라이언트/옵저버과, 클라이언트의 위치 및 상태와 채팅 내역을 저장할 데이터베이스, 데이터 처리를 담당하는 서버로 이루어진다.

2. System context diagram



(그림 4) System context diagram

- Connect : 접속
- Connect Notice : 접속 알림
- Move : 이동
- Is Danger : 위험 감지
- Send Message : 메시지 송신
- Reception Message : 메시지 수신
- Send SOS : SOS 요청
- Reception SOS : SOS 요청 확인
- Exit : 접속 종료
- Exit Notice : 접속 종료 알림
- Monitoring Clients : 클라이언트들의 위치 및 상태 모니터링
- Client Route : 클라이언트의 이동경로 확인
- Danger Client : 클라이언트가 위험 위치에서 오래 누적 시 알림
- Warning to Client : 클라이언트에게 위험위치 알림
- Help Client : 다른 클라이언트에게 도움 요청
- Get Location & State : 클라이언트가 자신의 위치와 상태를 확인

3. Use case list

3.1. 시스템 접속

Actor	Client
Description	클라이언트가 시스템에 접속한다. 접속을 하면 클라이언트는 ID를 배정받는다.

3.2. 시스템 접속 알림

Actor	Observer
Description	다른 클라이언트가 시스템에 접속된 것을 알려주고 해당 클라이언트를 모니터링 할 수 있게 한다.

3.3. 이동 및 위험감지

Actor	Client
Description	클라이언트가 가상 환경에서 움직이며 시스템은 해당 위치의 위험을 감지한다.

3.4. 클라이언트 위험 경고

Actor	Observer
Description	클라이언트가 위험 지역에서 오래 머물시 옵저버에게 알림이 간다.

3.5. 클라이언트 현재 상태 및 위치 확인

Actor	Client
Description	클라이언트가 자신의 현재 상태와 현재 위치(x,y좌표)를 확인한다.

3.6. 채팅

Actor	Client
Description	클라이언트가 다른 클라이언트와 주파수 채널을 통해 채팅한다.

3.7. 위험상황 SOS 요청

Actor	Client
Description	클라이언트가 자신이 위험한 상태에 놓여있어 시스템에 접속하고 있는 모두에게 SOS 도움 요청을 보낸다.

3.8. 시스템 접속 종료

Actor	Client
Description	다른 클라이언트가 시스템에 접속 종료한다.

3.9. 시스템 접속 종료 알림

Actor	Observer
Description	다른 클라이언트가 시스템에 접속 종료한 것을 알려주고 해당 클라이언트를 모니터링에서 제외시켜준다.

3.10. 클라이언트들의 위치 및 상태 모니터링

Actor	Observer
Description	클라이언트들의 위치 및 상태를 서버로 클라이언트가 보내주면 서버는 이를 종합한 정보를 정리해서 실시간으로 옵저버에서 확인할 수 있게 해준다.

3.11. 클라이언트의 이동 경로 확인

Actor	Observer
Description	옵저버에서 클라이언트의 이동경로를 확인 할 수 있다.

3.12. 클라이언트들에게 특정 위치 위험 알림

Actor	Observer
Description	옵저버에서 클라이언트들에게 특정 위치가 위험함을 알려준다.

3.13. 클라이언트 구조 요청

Actor	Observer
Description	옵저버에서 클라이언트의 움직임이 이상하거나 클라이언트가 위험해보일 때 다른 클라이언트들에게 해당 클라이언트의 위치를 전송하여 도움을 요청한다.

4. Concept of operation

4.1. 시스템 접속

Purpose	클라이언트가 시스템에 접속.
Approach	클라이언트가 클라이언트 프로그램에 접속하면 서버에게 ID를 배정받고 서버와 통신할 수 있게 연결된다. 그 후 서버에게 기존 클라이언트들의 리스트를 얻어 기존 클라이언트들과 채팅할 수 있게 메뉴가 추가되고 클라이언트는 가상의 재난 환경이 보이게 된다.
Dynamics	클라이언트가 시스템에 접속하고 싶은 경우.
Goals	시스템에 접속한 클라이언트에게 프로그램을 사용할 수 있게 한다.

4.2. 시스템 접속 알림

Purpose	다른 클라이언트가 시스템에 접속된 것을 알림.
Approach	시스템에 다른 클라이언트가 접속하면 옵저버에게 알림이 간다. 이후 옵저버는 해당 클라이언트를 모니터링 할 수 있다.
Dynamics	새로운 클라이언트가 시스템에 접속한 경우.
Goals	시스템에 새로 접속한 클라이언트를 알리며 옵저버에게는 모니터링 할 수 있게 한다.

4.3. 이동 및 위험감지

Purpose	클라이언트의 이동 및 위험 위치 감지.
Approach	클라이언트가 가상 환경에서 키보드를 통해 움직이며 클라이언트 프로그램은 해당 위치의 위험을 감지하며 클라이언트에게 알린다.
Dynamics	클라이언트가 키보드를 통해 움직였을 경우.
Goals	클라이언트가 움직이며 위험한 장소를 감지한다.

4.4. 클라이언트 위험 경고

Purpose	클라이언트의 위험 상태가 오래 지속될 때 옵저버에게 경고
Approach	클라이언트가 위험지역에서 너무 오래 머물고있으면 시스템은 옵저버에게 해당 클라이언트가 위험지역에 오래 머물고 있다고 경고한다.
Dynamics	클라이언트가 위험지역에 오래 머물 경우
Goals	클라이언트의 위험 상태가 오래 지속되면 옵저버가 빠르게 알 수 있게 한다.

4.5. 클라이언트 현재 상태 및 위치 확인

Purpose	클라이언트가 자신의 위치와 상태 확인
Approach	클라이언트가 자신의 위치와 상태를 알고싶을 때 x,y좌표 형태의 자신의 위치 상태정보를 확인한다.
Dynamics	클라이언트가 자신의 x,y좌표 형태의 위치와 상태 정보를 얻고 싶은 경우.
Goals	클라이언트가 x,y좌표 형태의 자신의 위치 상태정보를 확인할 수 있게 한다.

4.6. 채팅

Purpose	클라이언트가 다른 클라이언트와 주파수 채널을 통한 채팅.
Approach	클라이언트가 주파수 채널에 채팅을 보낼 때 서버로 양식에 맞게 전송하여 서버는 받은 데이터를 통해 다른 클라이언트에게 해당 채널로 전송한다. 이때 다른 클라이언트는 채팅 정보를 수신하여 띄워준다.
Dynamics	클라이언트가 주파수 채널을 통해 다른 클라이언트에게 채팅을 할 경우.
Goals	클라이언트와 다른 클라이언트간의 채팅을 할 수 있게 한다.

4.7. 위험상황 SOS 요청

Purpose	클라이언트의 SOS 도움 요청
Approach	클라이언트가 SOS 도움 요청을 서버로 보낸다. 이때 서버는 시스템에 접속하고 있는 모두에게 해당 클라이언트의 위치와 상태를 포함해서 SOS 도움 요청을 보낸다. SOS 도움 요청을 받은 클라이언트와 옵저버에 알림이 도착한다.
Dynamics	자신이 위험한 상태에 놓여있어 도움이 필요한 경우.
Goals	클라이언트가 도움이 필요할 때 시스템에 접속하고 있는 모두에게 도움을 요청하며 도움을 받을 수 있게 한다.

4.8. 시스템 접속 종료

Purpose	클라이언트의 시스템 접속 종료
Approach	클라이언트가 시스템 접속 종료를 하면 서버와 통신하고 있던 연결들이 끊기게 된 후 시스템이 종료한다.
Dynamics	시스템에 접속 종료할 경우
Goals	시스템에 접속을 종료하며 클라이언트와 서버 간의 연결을 끊는다.

4.9. 시스템 접속 종료 알림

Purpose	다른 클라이언트가 시스템에 접속 종료한 것을 알림
Approach	클라이언트가 시스템에 접속 종료를 하면 옵저버에게 알린다. 알림 받은 옵저버는 해당 클라이언트의 모니터링을 못하게 된다.
Dynamics	클라이언트가 시스템에 접속 종료한 경우.
Goals	클라이언트가 접속 종료한 후 시스템은 해당 클라이언트가 없는 환경처럼 만들어주도록 한다.

4.10. 클라이언트들의 위치 및 상태 모니터링

Purpose	클라이언트들의 실시간 위치 및 상태 모니터링
Approach	클라이언트들이 자신의 위치 및 상태를 서버로 보내주면 서버는 모든 클라이언트들의 위치 및 상태를 종합해서 실시간으로 옵저버로 전송한다. 이때 옵저버는 클라이언트들의 실시간 위치 및 상태를 모니터링 할 수 있게 된다.
Dynamics	클라이언트들의 위치 및 상태를 확인하는 경우.
Goals	클라이언트들의 위치 및 상태를 실시간으로 모니터링 할 수 있도록 한다.

4.11. 클라이언트의 이동 경로 확인

Purpose	클라이언트의 이동경로를 확인
Approach	클라이언트들의 자신의 위치 및 상태를 서버로 보내주면 서버는 이 정보를 데이터베이스에 저장한다. 옵저버에서 서버에게 클라이언트의 이동경로를 요청하면 서버는 데이터베이스의 정보를 가져와 옵저버에게 보내준다. 이때 옵저버는 클라이언트의 이동경로를 확인할 수 있다.
Dynamics	클라이언트의 이동경로를 확인하는 경우.
Goals	클라이언트의 이동경로를 옵저버에서 확인할 수 있도록 한다.

4.12. 클라이언트들에게 특정 위치 위험 알림

Purpose	옵저버에서 클라이언트들에게 특정 위치가 위험함을 알림
Approach	옵저버에서 해당 위치에 대해 위험하다는 것을 판단할 경우 클라이언트들에게 그 위치가 위험함을 위치 정보를 서버로 전송한다. 이후 서버는 모든 클라이언트들에게 전달받은 위치와 함께 전송하고 클라이언트들은 해당 위치가 위험함을 알림받는다.
Dynamics	클라이언트들에게 특정 위치가 위험함을 알릴 경우.
Goals	클라이언트들에게 특정 위치가 위험함을 알려준다.

4.13. 클라이언트 구조 요청

Purpose	옵저버에서 특정 클라이언트에 대한 구조 요청
Approach	옵저버에서 특정 클라이언트가 위험해보일 때 서버로 구조 요청을 보낸다. 이때 서버는 다른 클라이언트들에게 해당 클라이언트의 위치 정보와 함께 도움을 요청한다. 이후 클라이언트들은 해당 클라이언트의 위치와 함께 구조 요청을 받는다.
Dynamics	옵저버에서 특정 클라이언트에 대한 구조가 필요하다고 느낄 경우
Goals	옵저버에서 특정 클라이언트의 구조가 필요할 경우 다른 클라이언트들에게 이를 전달받을 수 있게 한다.

5. Problem statement

5.1. Overview

Real Time Location System은 클라이언트의 위치와 상태를 실시간으로 모니터링 하는 것이 주 목적이다. 또한 SOS 요청, 클라이언트 구조 요청 등 실시간으로 시스템에 접속한 모두에게 도움을 요청하여 위험한 상황에 대비할 수 있는 것이 장점이다.

이 시스템은 다음과 같은 목적이 달성할 수 있어야 한다.

- 정확한 데이터 처리
- 실시간 데이터 처리

5.2. Problem definition

Real Time Location System에서 클라이언트와 옵저버가 서버와 통신하는 과정에서 직면할 수 있는 문제점 및 해결 방법은 다음과 같다.

5.2.1. Problem#1 protocol

클라이언트와 옵저버는 서버를 통해 통신하기 때문에 정확한 통신을 위해서는 서로 전송하는 데이터들의 프로토콜을 정하는 것은 필수적이다.

Real Time Location System에서는 빠르고 정확한 통신이 필요하기에 프로토콜은 다음과 같은 방식으로 정한다.

- 먼저 Real Time Location System 데이터인지를 판단하기 위해 데이터 양 끝에 데이터의 시작과 끝을 표시한다.
- 데이터에는 첫 번째에 어떤 종류의 프로토콜인지 표시하고 뒤에는 해당하는 프로토콜의 데이터 정보가 들어간다.

(그림 5) 프로토콜 예시

프로토콜을 정하고 통신함으로써 잘못된 데이터를 사용하거나 다른 클라이언트에게 잘못 데이터를 보내는 일을 방지할 수 있어 보다 정확하게 통신할 수 있다.

5.2.2. Problem#2 Real-time communication between server and user

사용자와 서버는 실시간으로 데이터를 주고받으며 통신한다. 그러므로 신속한 통신을 위해 서버와 사용자들의 프로그램에서 데이터를 받고 처리하는 쓰레드를 배정해주는 형식으로 해결한다.

- 클라이언트는 서버에서 오는 데이터를 받고 처리하는 쓰레드를 배정받는다.

- 옵저버는 서버에서 오는 데이터를 받고 처리하는 쓰레드를 배정받는다.
- 서버는 클라이언트와 옵저버에서 오는 데이터를 받고 처리하는 쓰레드를 각 클라이언트와 옵저버에게 배정해준다.

클라이언트와 옵저버는 서버에게 데이터를 처리해주는 쓰레드를 하나씩 배정받는다. 이렇게 함으로써 사용자들은 1:1로 통신하는 것과 같이 실시간으로 보다 빠르게 통신할 수 있다.

5.2.3. Problem#3 클라이언트 관리

여러 클라이언트들이 사용하는 만큼 사용자들을 보기 쉽게 정리해서 관리하는 것이 중요하다.

- 새로운 클라이언트가 접속한 후 서버에서 쓰레드를 배정해주면 이들을 한곳에서 클라이언트들을 관리한다.
- 접속 종료하는 경우 배정했던 쓰레드를 제거하고 옵저버와 다른 클라이언트들에게도 접속 종료했음을 알려야한다.
- 옵저버에서도 클라이언트들을 한곳에서 관리하여야 한다. 새로운 클라이언트가 접속하면 신속하게 옵저버에서 추가하고, 접속 종료하면 옵저버에서 제외해야한다.

6. Glossary

Term	Description
Real Time Location System	클라이언트들의 위치 및 상태를 실시간으로 모니터링하고 서로 통신할 수 있게 해주는 시스템.
클라이언트	가상 환경에서 움직이고 활동하는 사용자.
옵저버	클라이언트들을 실시간으로 모니터링 하는 사용자.
모니터링	클라이언트들의 위치 및 상태를 지속적으로 감시, 관찰하는 행위.
쓰레드	어떠한 프로그램 내에서, 특히 프로세스 내에서 실행되는 흐름의 단위.
프로토콜	Real Time Location System에서 사용되는 데이터의 규칙 체계.
이동	클라이언트가 키보드를 통해 움직이는 행위.
위험감지	클라이언트가 위험 지역에 진입했는지에 대한 감지.
상태	클라이언트 위험감지 장치의 상태
채팅	클라이언트끼리 서버를 통해 문자기반으로 통신하는 행위.

7. References

- 그림 1,2 – https://namu.wiki/w/%EC%86%8C%EB%B0%A9%EA%B4%80
- 그림 3 – https://www.urbanbrush.net/