

## Course: Software Engineering (71032-01/02)

- 과제: AI 기반 Mobile Robot Controller를 UML을 사용하여 모델링하고 구현
- 팀별 개발 (3명/팀):
  - 프로젝트 계획서 제출: 10월16일(월): 제출물: 프로젝트 계획서
  - 중간발표: 11월 6일(월), 5분/팀: 제출물: 산출물(분석, 설계, 구현, 운영), UML 파일(설계결과물), 발표자료(파워포인트 자료)
  - 최종발표: 12월 4일(월), 5분/팀, 제출물: 산출물(분석, 설계, 구현, 운영), UML 파일(설계결과물), 소스코드, 발표자료(파워포인트 자료)
  - 작성요령: 산출물(분석, 설계, 구현, 운영)에 UML 파일 내용을 복사하여 추가함.
  - 제출요령: 온라인강의실 강좌Q&A에 제출
  - 제목예: 팀명:프로젝트계획서제출, 팀명:중간발표제출, 팀명:최종발표제출

### << Mobile Robot Controller >>

화재, 지진의 발생, 폭발, 독성 물질 유출과 같은 여러 종류의 재난 상황이 발생한 경우에 체계적인 상황 해제를 위해서는, 재난 지역에서 위험 지점은 어디이고 구조할 대상이 있다면 탈출 경로는 무엇인지 등의 정보를 파악할 필요성이 있다. 특히 그 지역에서 사람이 활동하기 힘들다면, 사람 대신 mobile robot이 위의 작업을 수행할 수 있을지도 모른다.

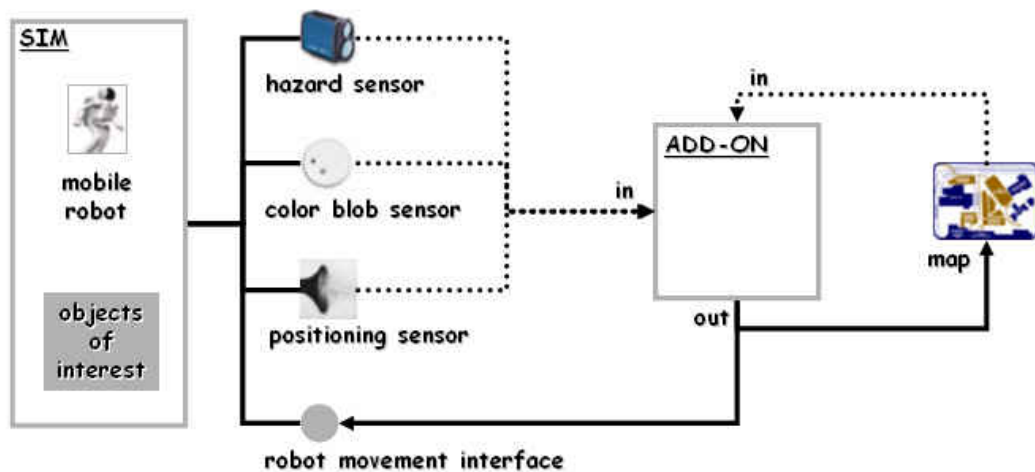
Guru-Guru Inc.라는 회사는 위와 같은 경우에 사용할 수 있는 mobile robot에 대한 일반적인 연구를 수행하기 위해서 ROBO lab.이라는 연구소를 세우고, 우선 mobile robot의 동작을 시뮬레이션 하는 시스템인 SIM (Mobile Robot Simulator)을 만들었다. 그런데 SIM은 오퍼레이터가 재난 지역을 완전히 알고 있는 상태에서 mobile robot의 robot movement interface를 통해서 로봇의 동작을 수동으로 제어하는 시스템이기 때문에, 재난 지역에 대한 정보가 부족하거나 재난 지역이 방대하면 사용하기가 힘들다. 이에 ROBO lab.에서는 SIM을 재사용하되 재난 지역에 대한 정보가 부족한 상태에서 로봇의 동작을 자동으로 제어할 수 있도록 하는 시스템인 ADD-ON (Additional Mobile Robot Controller)을 만들어서 SIM과 더불어 사용하려고 한다.

### >> System Boundary

ADD-ON은 SIM으로부터 세 개의 입력을 받는다.

ADD-ON은 SIM의 robot movement interface를 통해서 로봇의 동작을 지시한다.

ADD-ON은 지도를 입력 받고 경우에 따라 지도를 재구성한다.



지도는 재난 지역을 추상화한 모델이다. 지도는 2차원 좌표공간(단, x, y 좌표는 각각 0 또는 자연 수 값만 가질 수 있다.)으로 구성되며 각 좌표를 지점(spot)이라고 칭한다. 지도상의 한 지점에 위험물이 존재하면 그 지점은 위험 지점이라고 부르고, Color blob이 발견되면 그 지점은 중요 지점이라고 부른다. 지도는 불완전하다. 즉 지도상의 한 지점에 위험 지점이라는 표시가 없더라도 실제로는 위험 지점일 가능성이 있다. ADD-ON은 오퍼레이터로부터 지도를 얻는다.

hazard sensor는 로봇의 전방 1칸 앞이 위험 지점인지를 판별하는 센서이다.

color blob sensor는 로봇의 전후 좌우 1칸이 각각 중요 지점인지를 판별하는 센서이다.

positioning sensor는 로봇의 지도상 좌표 위치를 나타내주는 센서이다.

ADD-ON은 SIM으로의 호출을 통해서 센서들의 값을 얻을 수 있다고 가정한다.

## >> Requirements

ADD-ON 시스템은 다음의 요구사항을 만족시켜야 한다.

### **Handling map data:**

오퍼레이터로부터 지도 자료를 입력 받아 재난 지역 모델을 구성해야 한다.

재난 지역 모델이 변경되면 이를 지도에 반영한다.

(앞으로 지도와 재난 지역 모델은 같은 뜻으로 사용한다.)

### **Planning a path:**

오퍼레이터로부터 (로봇의 출발점, 로봇이 탐색할 지점들의 순서집합)을 입력 받아 로봇이 출발점에서 탐색할 지점들 사이를 이동할 경로를 만들어야 한다. 단, 이동 경로는 위험 지점을 포함하지 않아야 한다.

### **Following a given path:**

로봇이 정해진 경로를 따라 이동할 수 있도록 동작을 지시해야 한다. 이때 시스템은 SIM의 robot movement interface를 사용해야 한다. SIM의 robot movement interface는 로봇이 지도상에서 앞으로 1칸 이동, 시계 방향으로 90도 회전하는 기능을 제공한다고 가정한다.

### Avoiding hidden hazards:

지도, 즉 환경 정보는 완벽하지 않음을 가정한다. 예를 들어 지도상에서는 어떤 지점이 위험 지점이 아니라고 하지만 실제로는 위험 지점인 경우가 있다. SIM으로부터 hazard sensor 값을 입력 받아 특정 지점에서 숨어있는 위험물을 발견하면 그 지점을 위험 지점으로 표시한 후 새로운 경로를 계산한다.

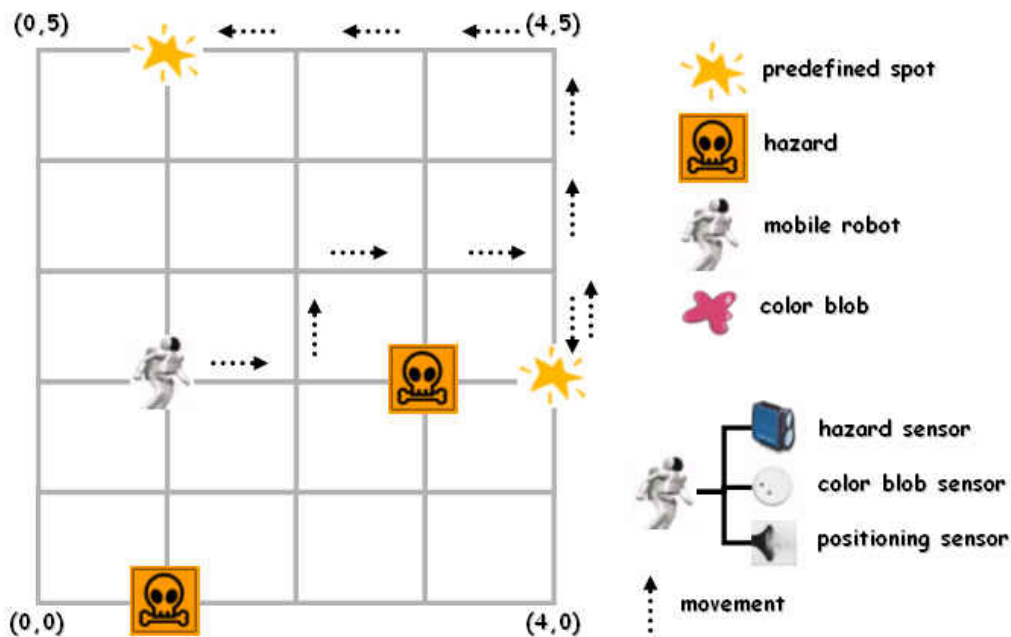
### Detecting color blobs (useful information):

지도상의 특정 지점에서 Color blob이 발견되면 그 지점은 중요 지점이라고 가정한다. SIM으로부터 color blob sensor 값을 입력 받아 특정 지점에서 Color blob을 발견하면 그 지점을 중요 지점으로 표시한다.

### Compensating for imperfect motion:

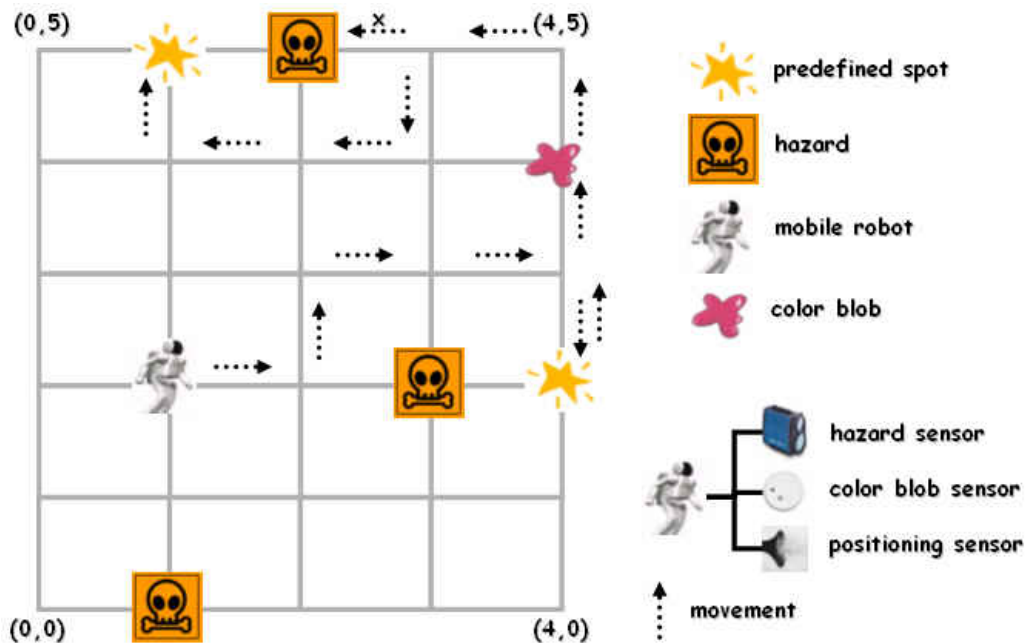
로봇의 동작은 지시받은 동작과 정확하게 일치하지 않을 가능성이 있다는 것을 가정한다. 즉 시스템이 SIM의 robot movement interface를 통해서 로봇을 앞으로 1칸 이동시켰을 때 로봇이 움직이지 않거나 예상보다 많은 2칸을 움직였을 가능성이 있다. SIM으로부터 positioning sensor 값을 입력 받아 위와 같은 경우가 발생했음을 파악할 수 있어야 하고, 그러한 경우에는 잘못된 동작을 보상하여 다음 이동에 영향이 없도록 하던지 로봇의 이동 경로를 새롭게 변경해야 한다.

## >> Typical Scenarios



<그림. 1>

<그림. 1>에서 로봇은 (1,2) 지점에서 출발하여 (4,2) 지점과 (1,5) 지점을 탐색하기 위해서 계산한 경로에 따라 이동한다. 이동 경로는 (1,0), (3,2)의 위험 지점을 포함하지 않는다.



<그림. 2>

<그림. 2>에서 로봇은 정해진 이동 경로를 따라 (4,2)를 탐색했고 (1,5)로 이동하는 도중에 (4,4)에서 color blob을 발견했다. 이후의 경로에서 (2,5)가 지도와는 달리 위험 지점임을 발견하고 새로 경로를 계산하여 (1,5)를 탐색한다.

### 1. 입력형식:

Map: (4 5) // 지도 크기

Start: (1 2) // 시작 위치

Spot: ((4 2)(1 5)) // 탐색 위치

Hazard: ((1 0)(3 2)) // 위험 지점

- 프로그램 실행 도중에 일시 정지하고 음성인식으로 중요(color blob)/위험(hazard) 지점 생성함

### 2. 출력 결과

- 지도를 표시하고 로봇과 미리 정해진 탐색위치, 위험지점 표시
- 탐색 위치를 향해 로봇이 이동하는 모습을 표시하고 도중에 중요/위험 지점을 생성하고 발견하여 다시 경로 계산하여 이동
- 최종적으로 모든 탐색 위치를 방문