

객체지향개발방법론 Practice #2

202211287 김태인

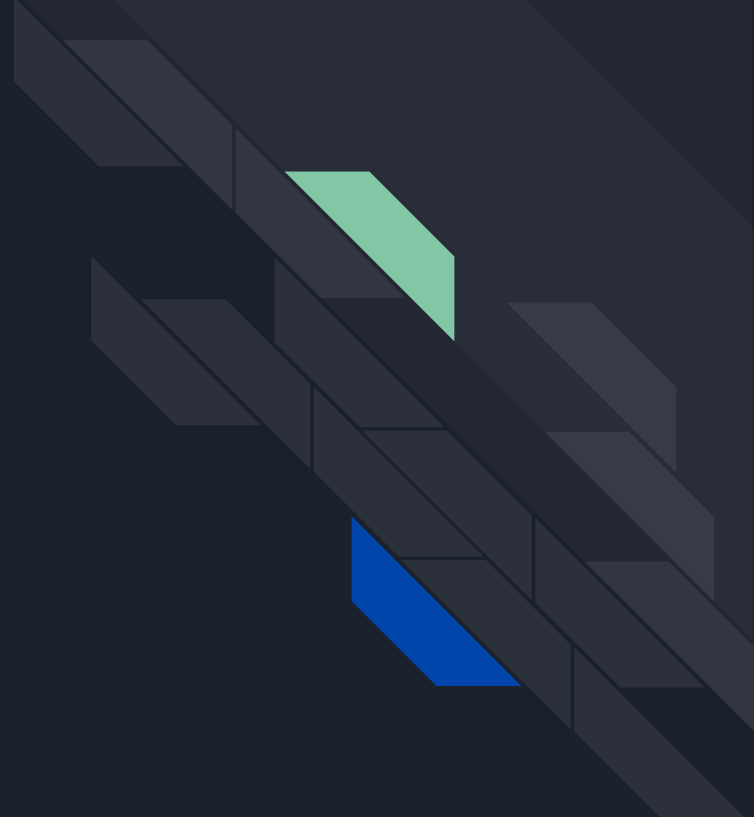
202311252 곽수호

202111368 정선민

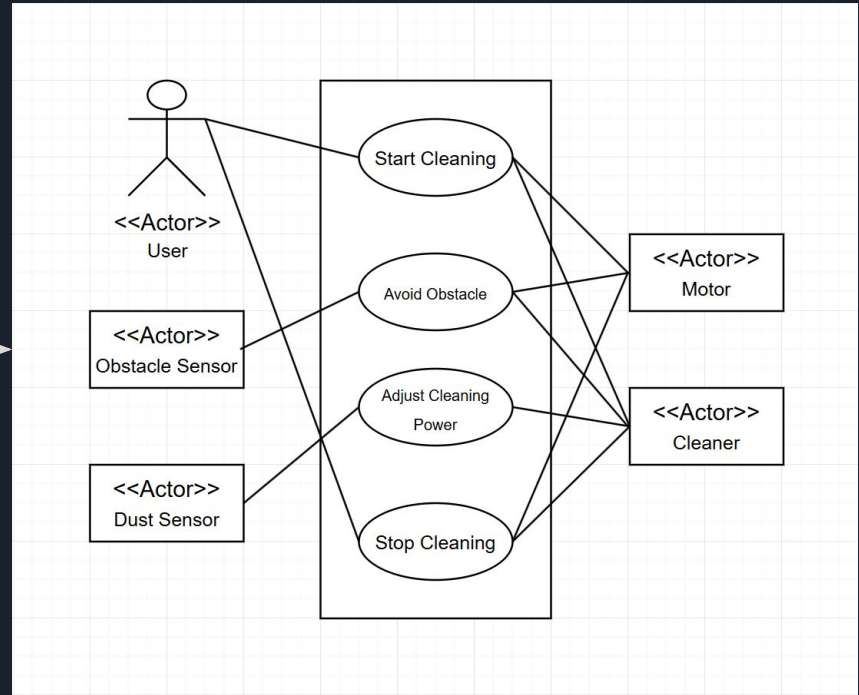
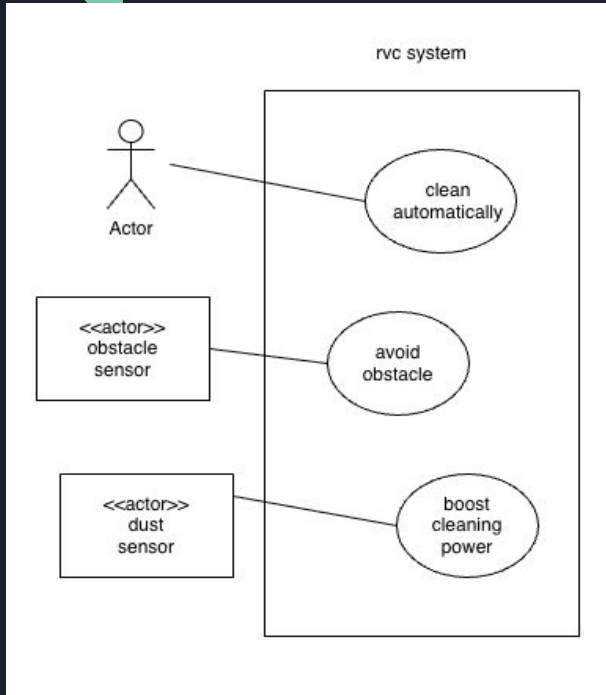
202211378 조성원

목차

1. (Refined)Use case
2. System sequence diagram
3. Domain model



Use case Diagram





Use case

1

Use Case : 1. Start Cleaning

Actors : User, Motor, Cleaner

Description :

- 사용자가 청소를 시작(ex. 시작버튼 누름, 모바일 구동)하면 Use case가 시작됨
- 시스템에서 일반 청소기능을 활성화하고 로봇에게 전진을 명령함
- 시스템은 청소 과정 동안 센서 입력(장애물, 먼지)를 지속적으로 모니터링함



Use case

1

Pre-requisites :

- 모터가 제어 가능한 상태여야 한다.
- 클리너가 제어 가능한 상태여야 한다.

Typical Courses of Events : (S) System (A) Actor

1. (A) 사용자가 청소를 시작(ex. 시작버튼 누름, 모바일 구동)한다
2. (S) 시스템이 **Obstacle Sensor, Dust Sensor**의 최초 입력을 받는다
3. (S) 전방장애물이 감지되지 않는 경우 전진 이동과 클리너 일반 작동을 시작한다
4. 유즈케이스가 종료된다



Use case

1

Alternative Courses of Events :

- A1. 시스템이 전방 장애물을 감지한 경우
 - 2. Avoid Obstacle 유스케이스가 시작된다
 - 이후 기본 흐름 3단계로 진행한다

Exceptional Courses of Events :

- 시스템이 Obstacle Sensor, Dust Sensor 중 어느 하나의 최초 입력을 받지 못한 경우
 - 해당 센서의 오류가 있음으로 판단하여 시스템은 로봇을 즉각 정지



Use case

2

Use Case : 2. Avoid Obstacle

Actors : Obstacle Sensor, Motor, Cleaner

Description :

- 청소 중 장애물 센서가 전방 장애물을 감지하면 이 유스케이스가 시작됨.
- 시스템은 로봇의 동작을 중지시킴
- 시스템은 장애물 위치에 따라 회피 기동(좌/우 회전)을 수행하며, 사방이 막힌 경우 후진 후 경로를 재설정함.
- 로봇이 장애물을 피해 전진 이동을 재개하면 유스케이스가 종료됨.



Use case

2

Pre-requisites :

- 로봇이 청소 동작을 수행하며 전진중이어야 한다.
- 장애물 센서가 정상적으로 동작하고 있어야 한다.
- 모터가 제어 가능한 상태여야 한다.
- 클리너가 제어 가능한 상태여야 한다.

Typical Courses of Events : (S) System (A) Actor

1. (A) Obstacle Sensor가 전방 장애물을 감지한다.
2. (A) 센서는 장애물 감지 신호를 시스템에 전달한다.
3. (S) 시스템은 현재 이동과 클리너에게 정지명령을 전달한다.
4. (A) 모터와 클리너가 정지한다.
5. (S) 시스템은 회피 가능한 방향을 판단한다.
6. (S) 시스템은 좌측이 비어 있음을 확인한다.
7. (S) 시스템은 모터에 좌회전 명령을 전달한다.
8. (A) 모터는 좌회전하여 장애물을 회피한다.
9. (S) 시스템은 새로운 주행 가능 경로를 확인한다.
10. (S) 시스템은 전진 이동과 클리너 시작을 전달한다
11. (A) 클리너를 키고 모터는 전진한다.
12. 유스케이스가 종료된다.



Use case

2

Alternative Courses of Events :

- A1. 오른쪽으로 회피 가능한 경우
 - 5a. 시스템은 좌측에 장애물이 있고 우측은 비어 있음을 확인한다.
 - 6a. 시스템은 모터에 우회전 명령을 전달한다.
 - 7a. 모터는 우회전하여 장애물을 회피한다.
 - 이후 기본 흐름 9단계로 진행한다
- A2. 양쪽 모두 장애물이 있는 경우
 - 5b. 시스템은 좌우 양쪽 모두 장애물이 있음을 확인한다.
 - 6b. 시스템은 모터에 후진 명령을 전달한다.
 - 7b. 모터는 양옆 중 하나라도 장애물이 없는 지점까지 후진한다.
 - 8b. 시스템은 다시 회피 가능한 방향을 판단한다.
 - 9b. 시스템은 좌회전(우선) 또는 우회전 명령을 전달한다.
 - 이후 클리너 on 전진 이동을 재개한다.



Use case

2

Exceptional Courses of Events :

- 장애물 감지 센서 오류
 - 좌회전 명령 또는 우회전 명령을 받고 로봇을 회전시켰을 때 회전시킨 방향의 반대 센서가 장애물을 없다고 인지하는 경우 장애물 감지 센서 3개중 최소 하나는 오류가 있음으로 판단하여 시스템은 로봇을 즉각 정지.



Use case

3

Use Case : 3. Adjust Cleaning Power

Actors : Dust Sensor, Cleaner

Description :

- 청소 중 먼지 센서가 먼지를 감지하면 이 유스케이스가 시작됨.
- 시스템은 사전에 정의된 시간 동안 흡입력을 높여 집중 청소(Boost)를 수행함.
- 설정된 시간이 경과하면 시스템은 흡입력을 다시 일반 수준으로 되돌림.
- 흡입력이 정상화되거나, 장애물 감지 등으로 인해 (2. Avoid Obstacle)이 시작될 경우, 또는 사용자에 의해 청소가 중단되면 유스케이스가 종료됨.



Use case

3

Pre-requisites :

- 로봇이 청소 동작을 수행하며 전진중이어야 한다.
- 먼지 센서가 정상적으로 동작하고 있어야 한다.
- 클리너가 제어 가능한 상태여야 한다.

Typical Courses of Events : (S) System (A) Actor

1. (A) Dust Sensor가 전방 먼지를 감지하여 시스템에 전달한다.
2. (S) 먼지 감지 시 시스템은 클리너에 흡입력 증가 명령을 전달한다.
3. (A) 클리너가 흡입력을 증가시킨다.
4. (S) 시스템은 일정시간 후(ex10초) 클리너에게 흡입력 정상 명령을 전달한다.
5. (A) 클리너가 흡입력을 정상 단계로 내린다.
6. 유스케이스가 종료된다.



Use case

3

Alternative Courses of Events :

- A1. 흡입력 증가가 끝나기 전에 Dust Sensor가 먼지를 또 감지한 경우
 - 5. 시스템은 먼지를 다시 감지한 시점부터 일정시간(ex10초)후에 클리너에게 흡입력 정상 명령을 전달한다.
 - 이후 기본 흐름 5단계로 진행한다
- A2. Use Case가 끝나기 전 2. Avoid Obstacle 유스케이스가 시작된 경우
 - Use Case가 종료된다

Exceptional Courses of Events :

N/A



Use case

4

Use Case : 4. Stop Cleaning

Actors : User, Motor, Cleaner

Description :

- 사용자가 청소를 중단(ex. 중단버튼 누름, 모바일 구동)하면 Use case가 시작됨
- 시스템에서 일반 청소기능을 비활성화하고 로봇에게 모터 중지를 명령
- 시스템은 센서(장애물, 먼지)를 비활성화함
- 유스케이스가 종료됨.



Use case

4

Pre-requisites :

- 모터가 제어 가능한 상태여야 한다.
- 클리너가 제어 가능한 상태여야 한다.

Typical Courses of Events :

1. (A) 사용자가 청소를 중단(ex. 중단버튼 누름, 모바일 구동)한다
2. (S) 시스템은 모터와 클리너의 동작을 중지한다.
3. (S) 시스템이 Obstacle Sensor, Dust Sensor의 연결을 해제한다
4. 유즈케이스가 종료된다

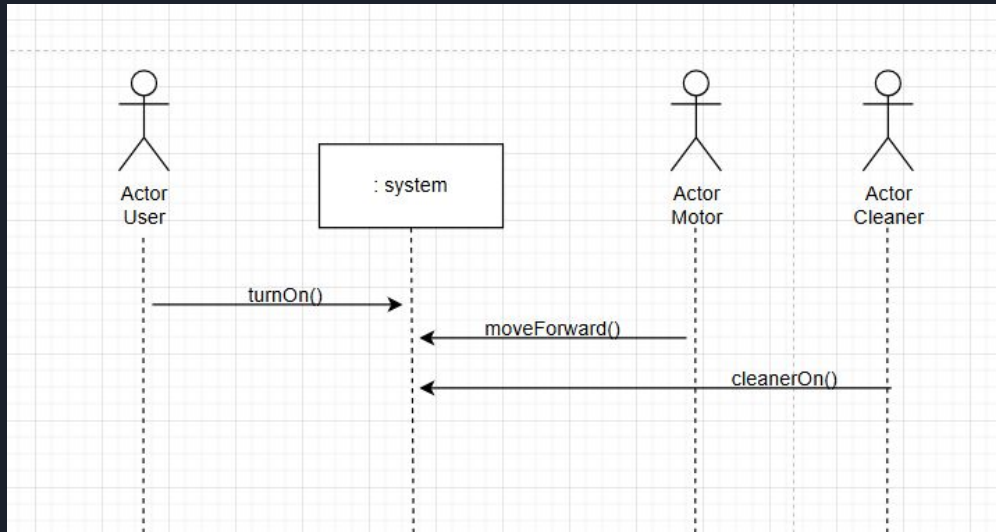
Alternative Courses of Events :

- N/A

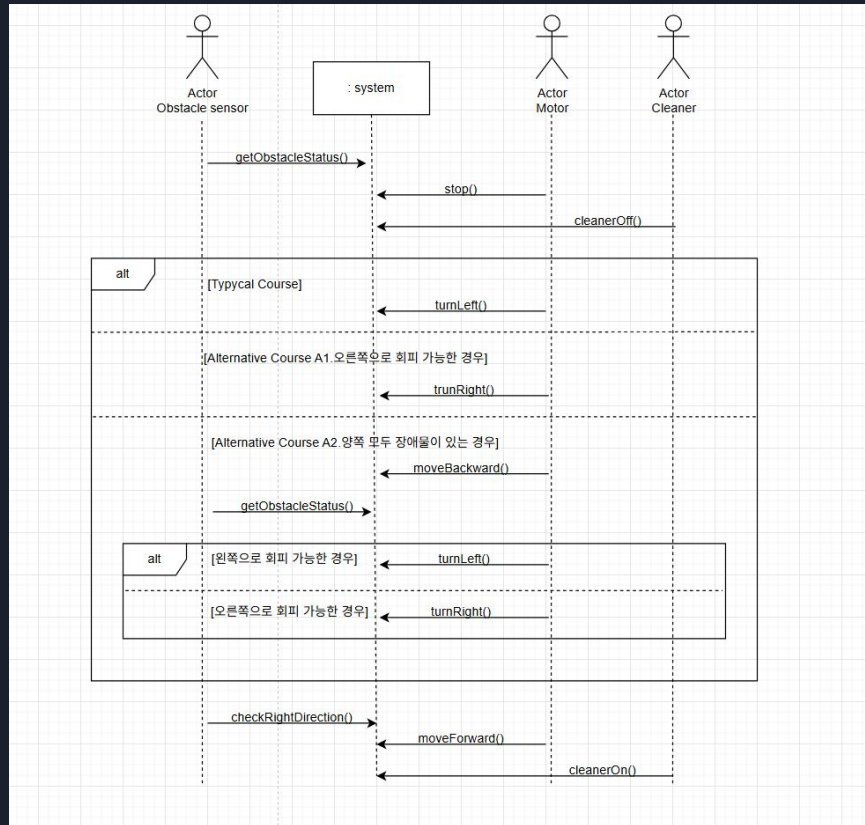
Exceptional Courses of Events :

- N/A

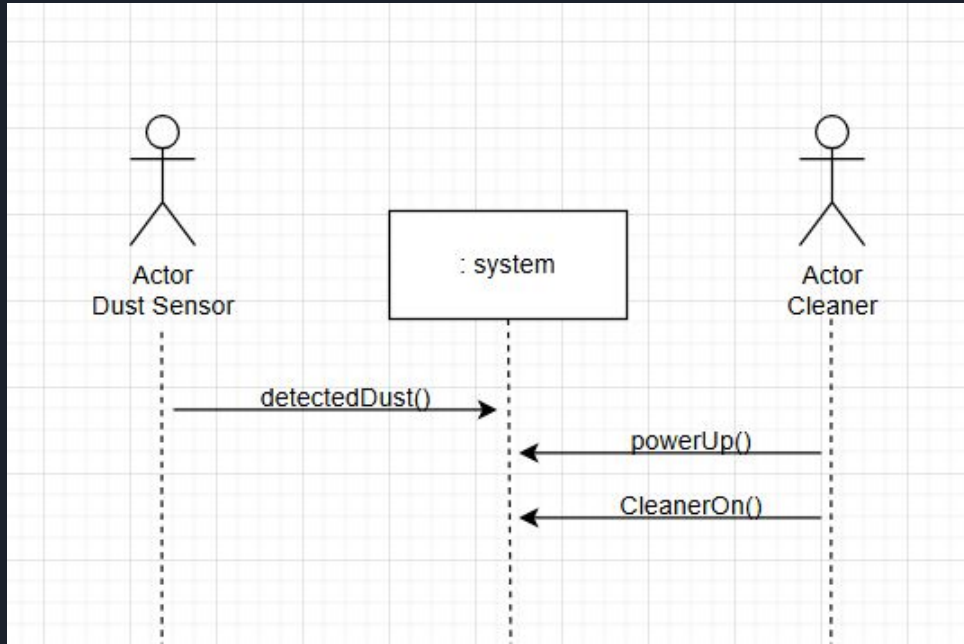
System Sequence diagram #1



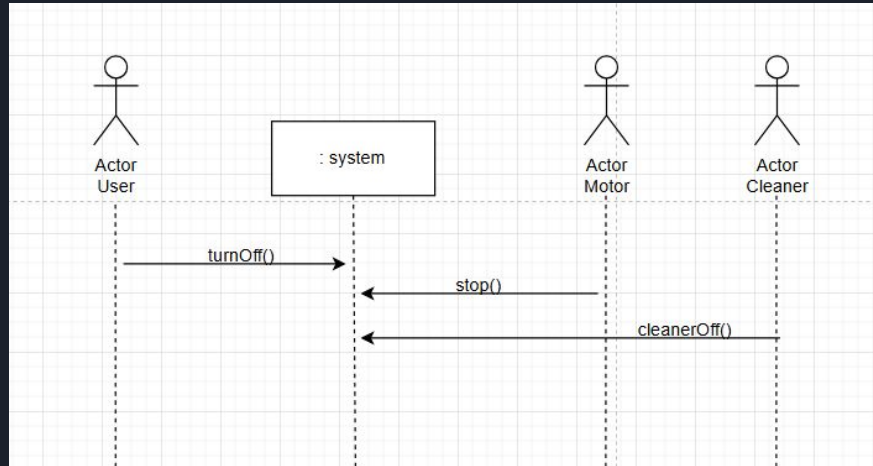
System Sequence diagram #2



System Sequence diagram #3



System Sequence diagram #4



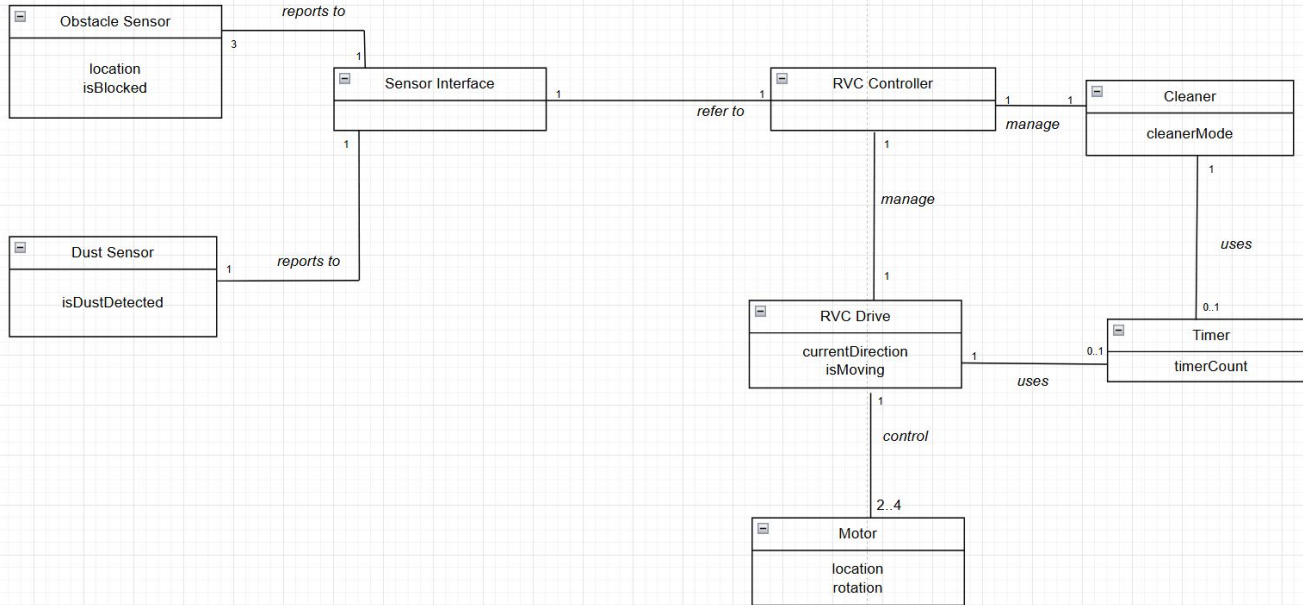


System Operations (System Interface)

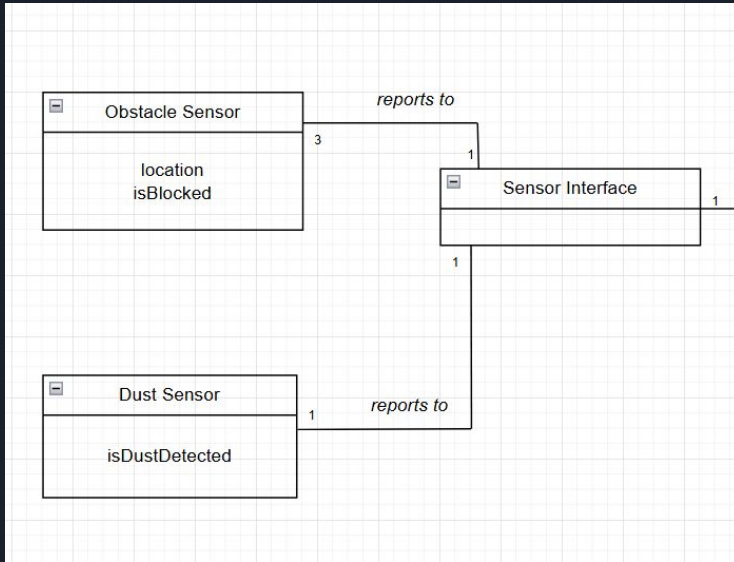
RVC system

- + turnOn()
- + turnOff()
- + getObstacleStatus()
- + moveForward()
- + turnLeft()
- + turnRight()
- + moveBackward()
- + stop()
- + checkRightDirection()
- + CleanerOn()
- + CleanerOff()
- + PowerUp()
- + DetectedDust()

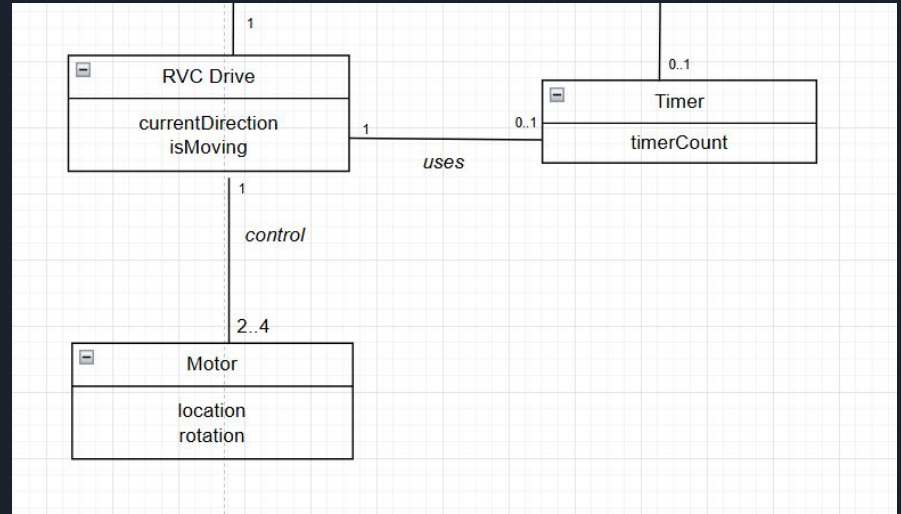
Domain model



Domain model

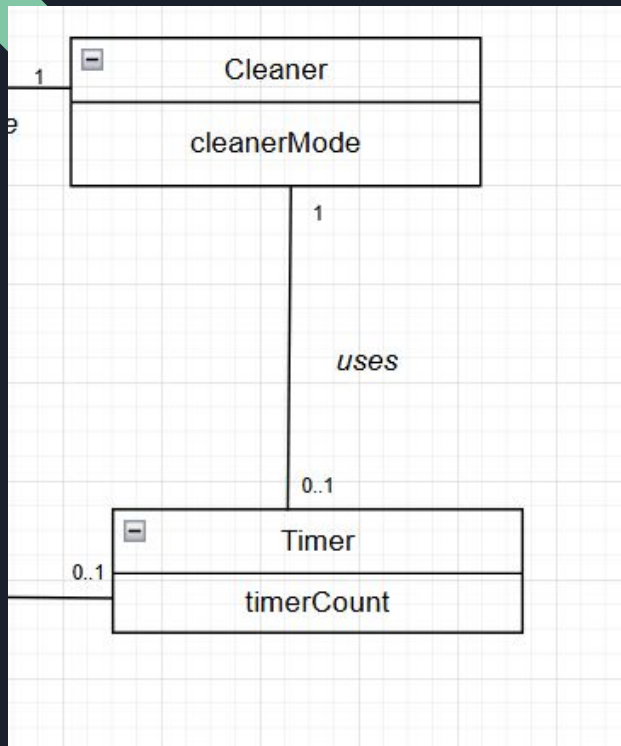


인식 시스템 객체

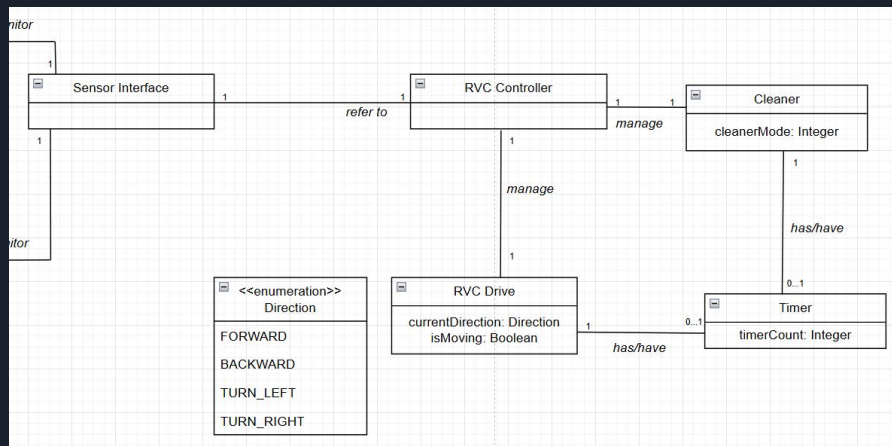


모터구동 객체

Domain model



클리너 객체



컨트롤러 객체



감사합니다.