

객체지향개발방법론

TEAM3 발표

202213351 김태성

202111382 최성준

202011380 최용근

202011434 최원탁

Contents.

01

UseCase Diagram Refine

02

Domain Model Refine

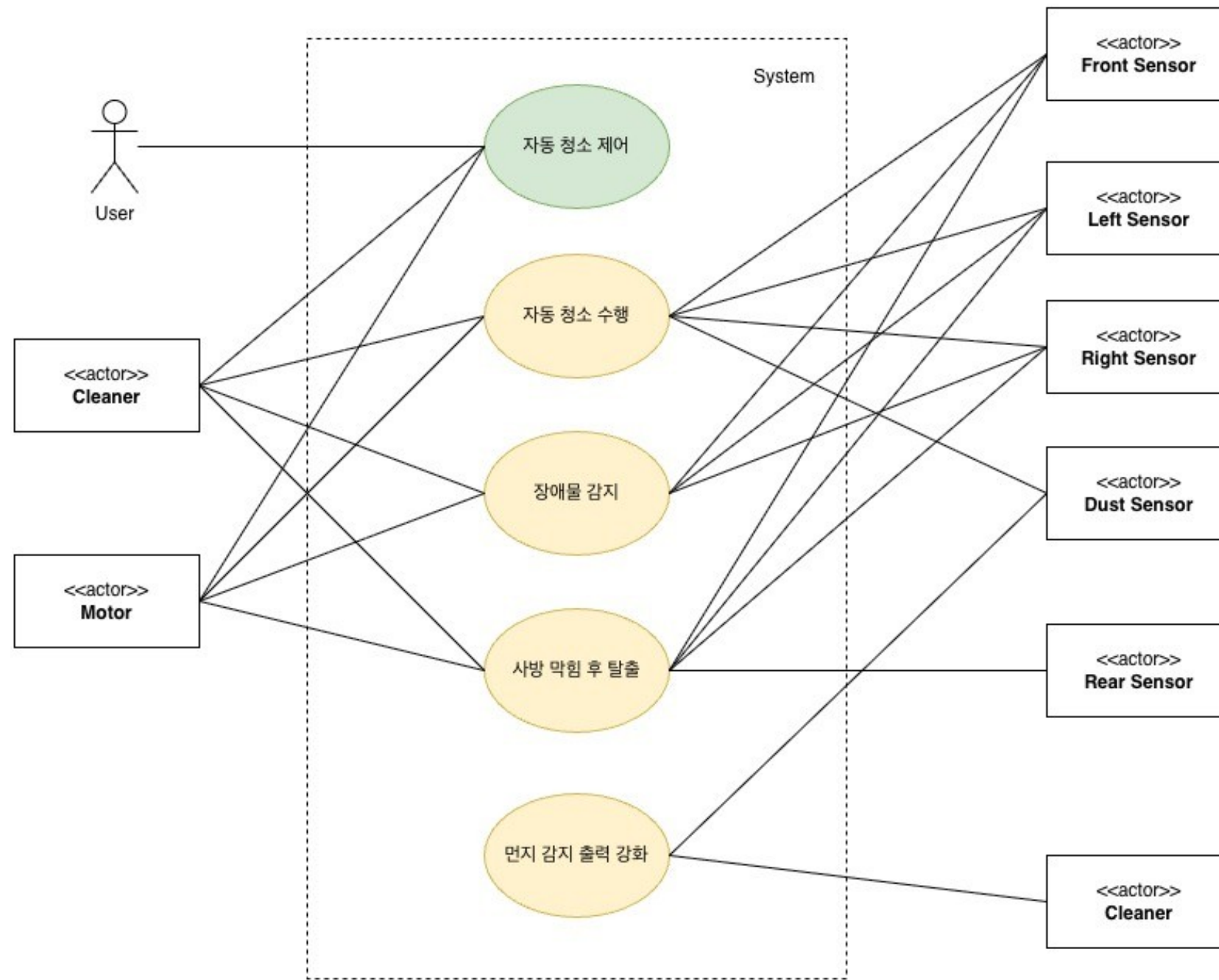
03

Sequence Diagram

04

Class Diagram

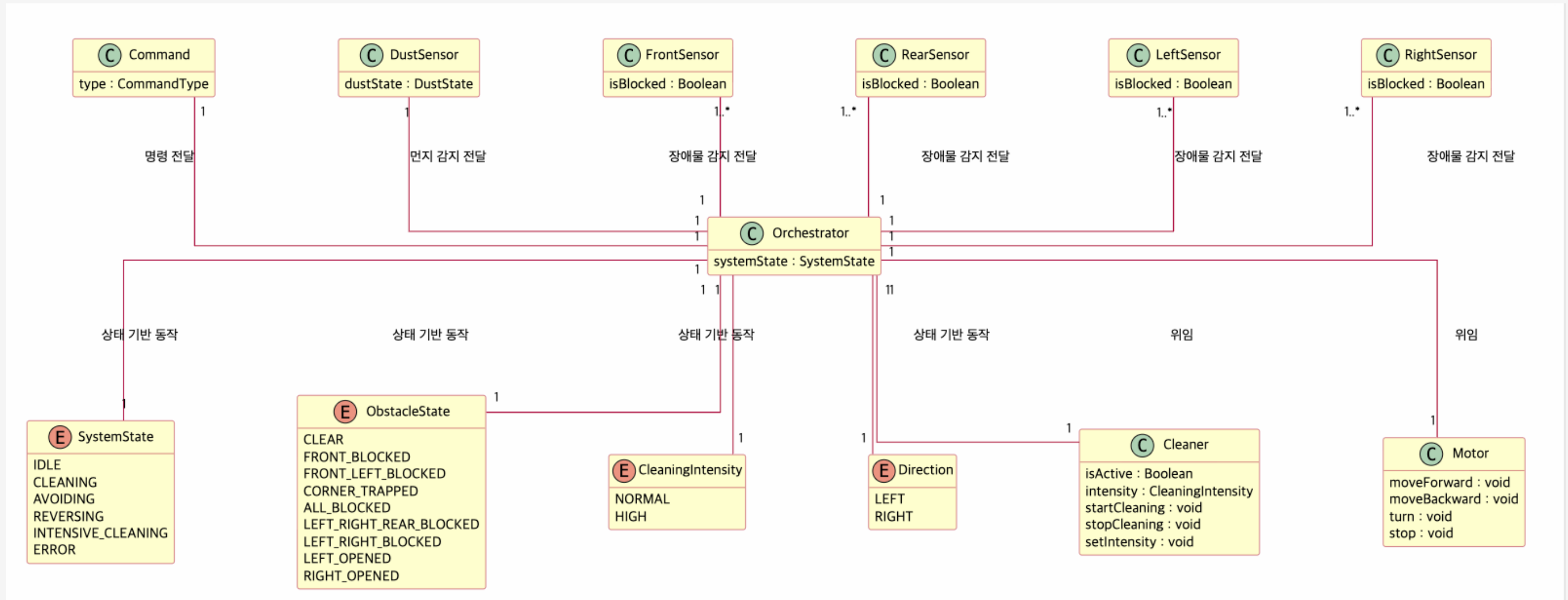
UseCase Diagram Refine



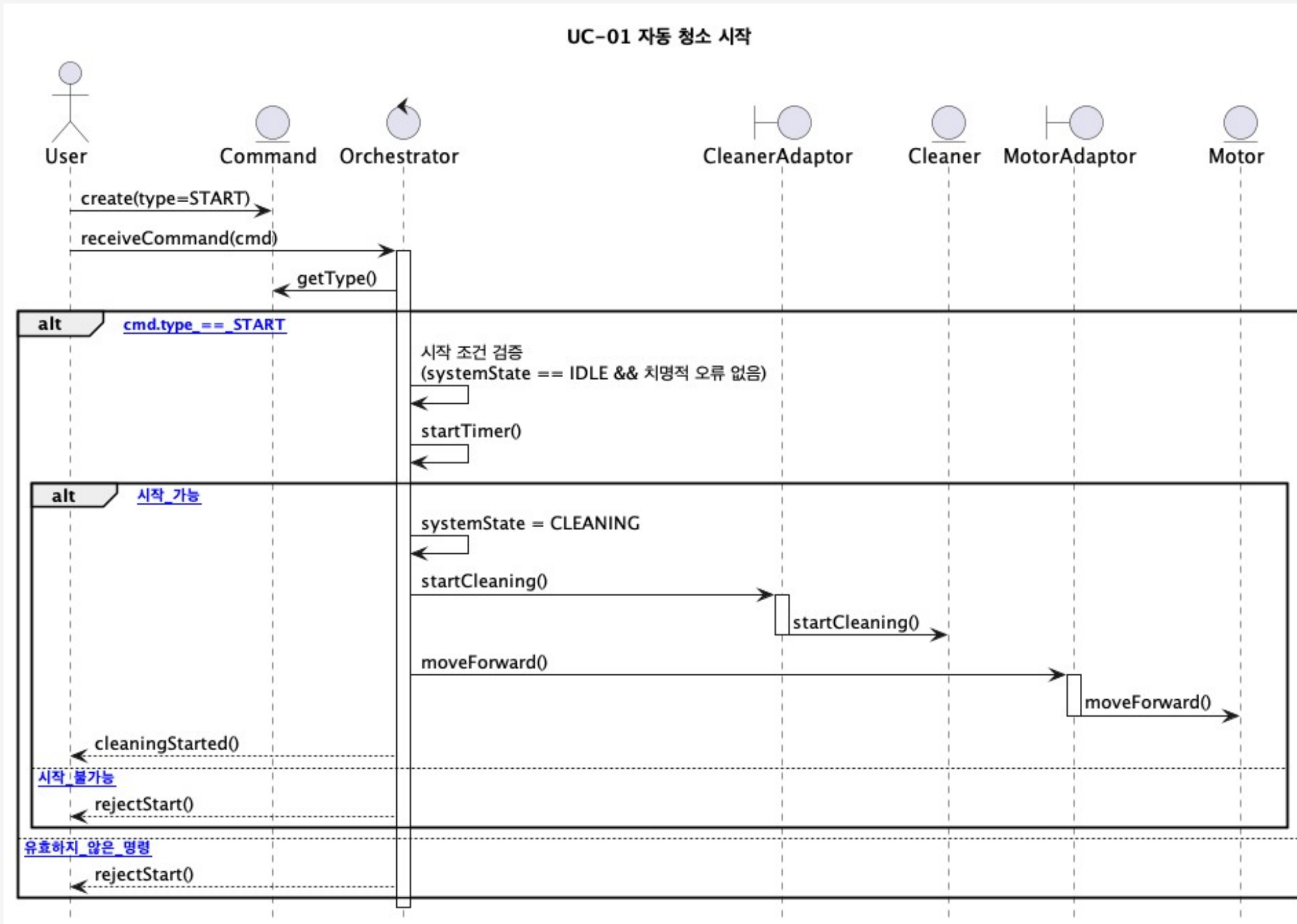
- **Obstacle Sensors**
→ **Front, Left, Right Sensor**
세분화

Domain Model Refine

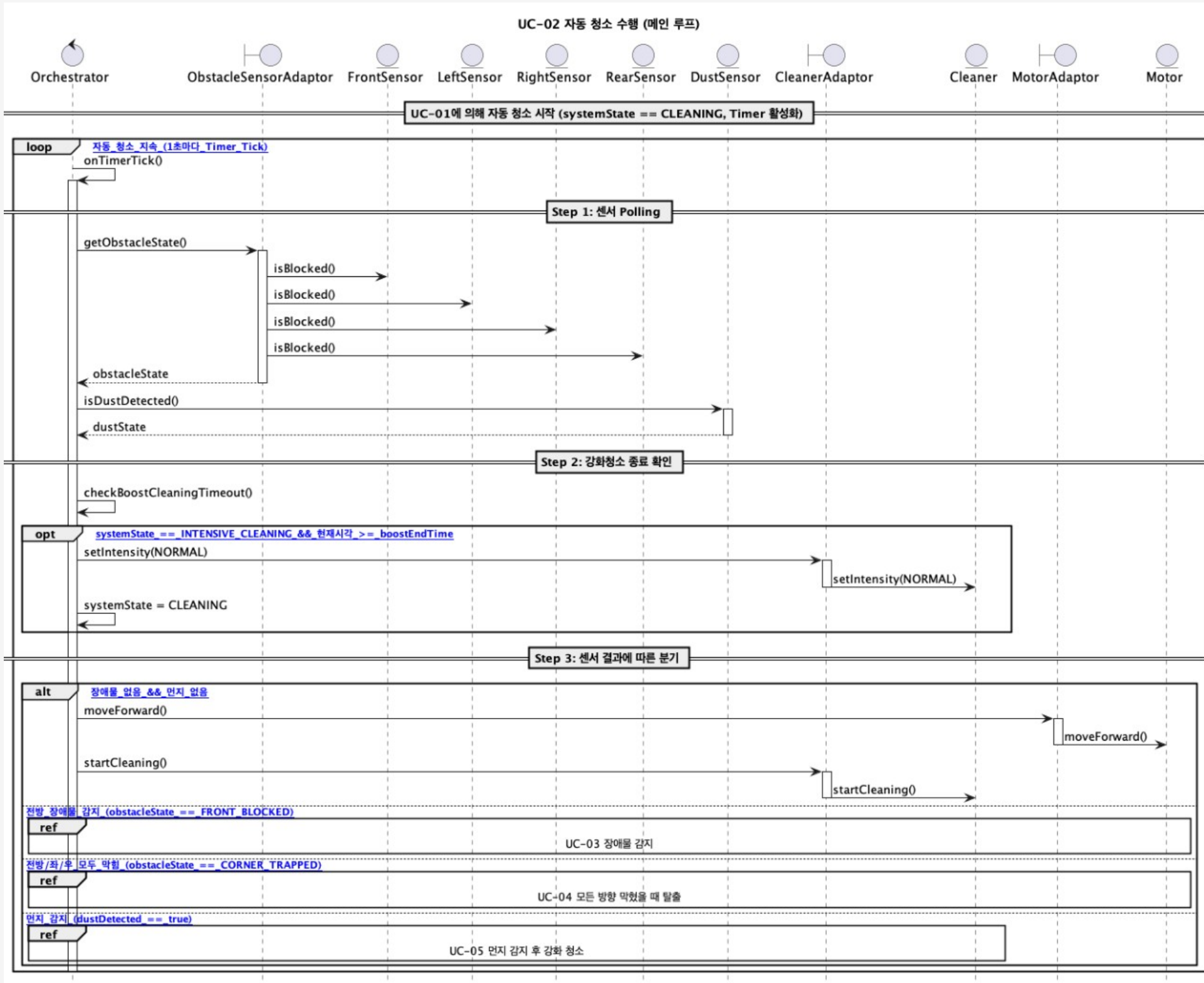
- Adaptor 제거
- 상태 추가



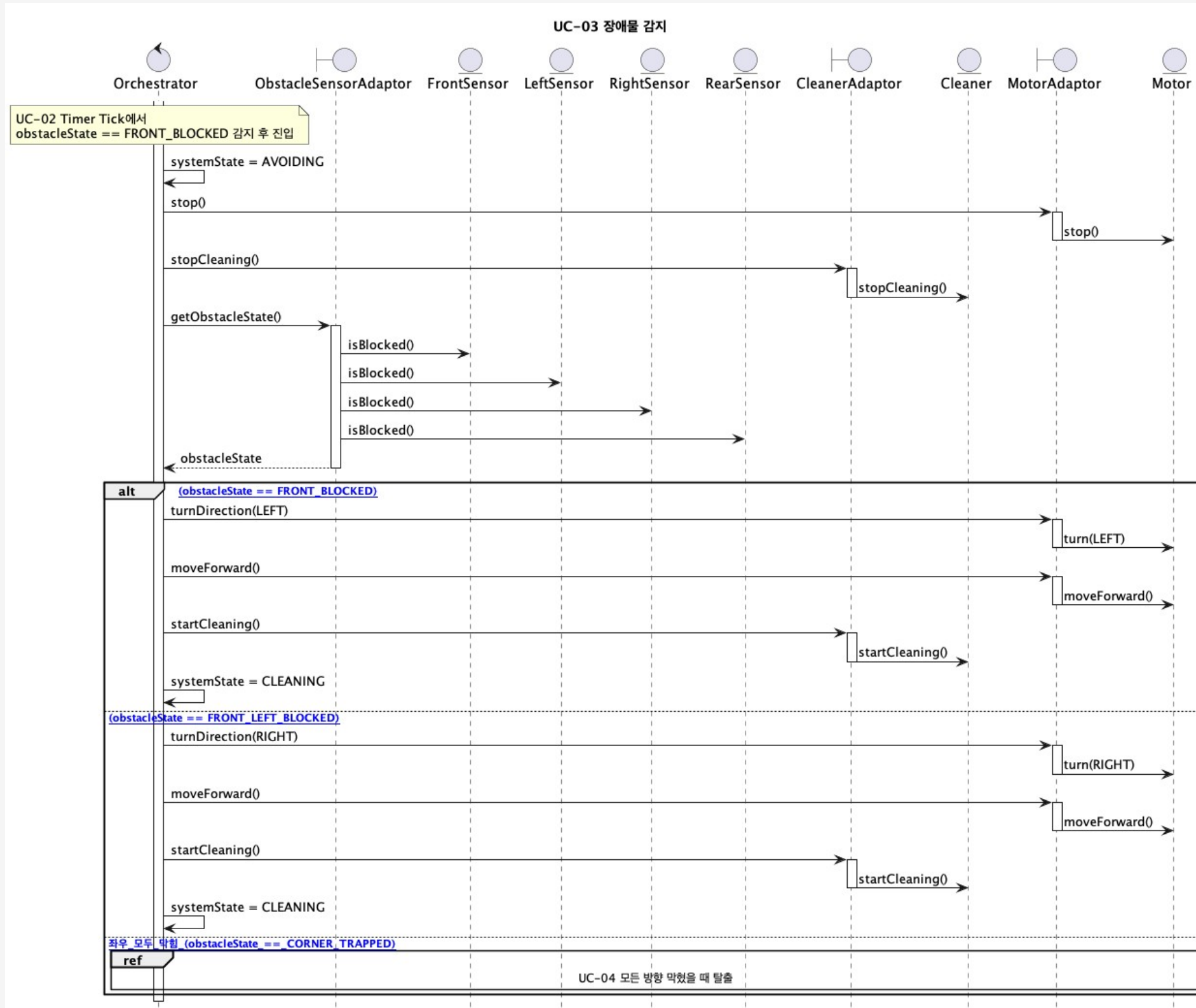
Sequence Diagram



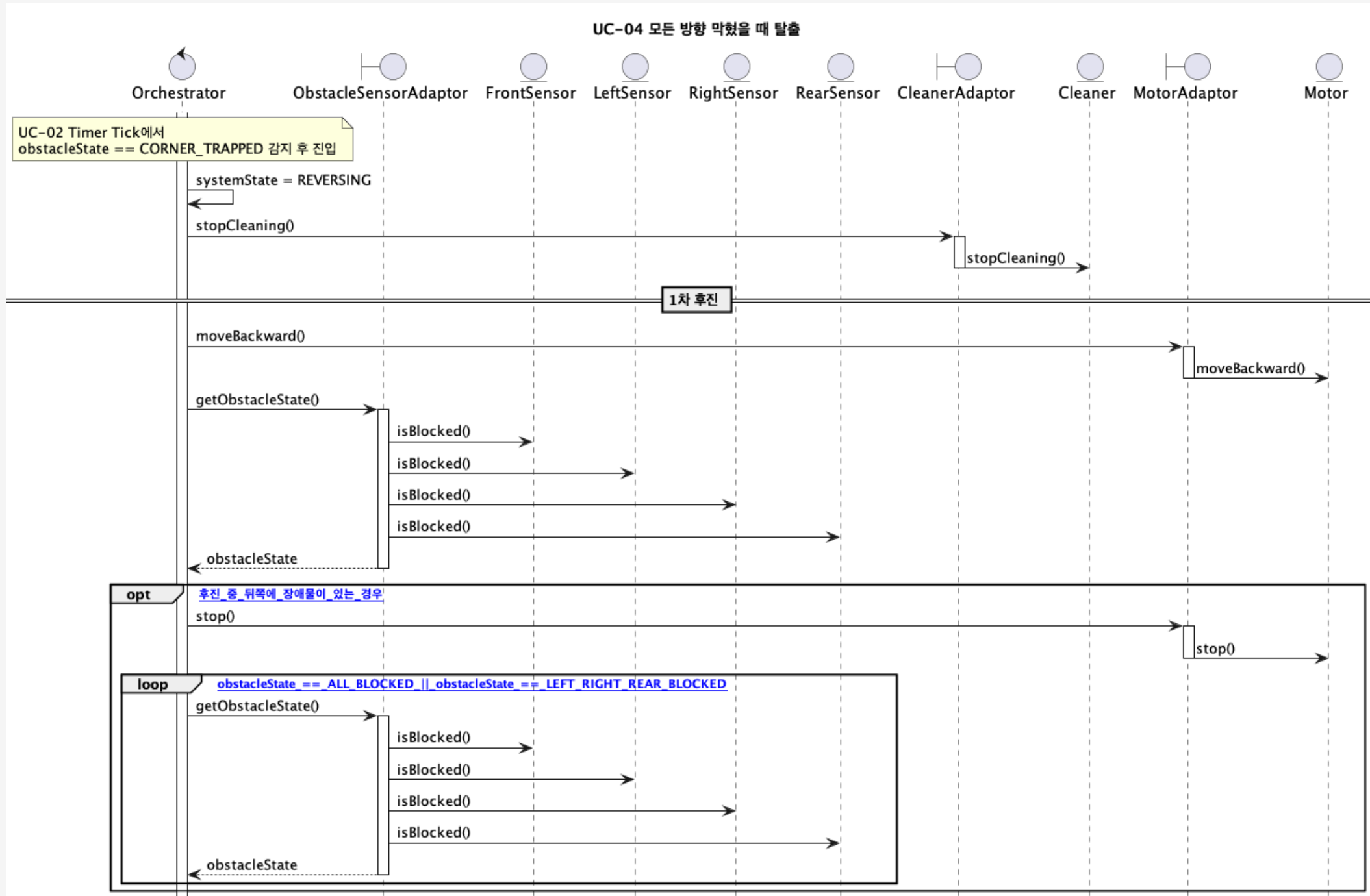
- 사용자의 Command로 Trigger
- Orchestrator가 시작 가능한 상황인지를 검증
- Polling을 위한 Timer Start
- 검증 성공 시 Cleaner와 Motor에 실행 명령



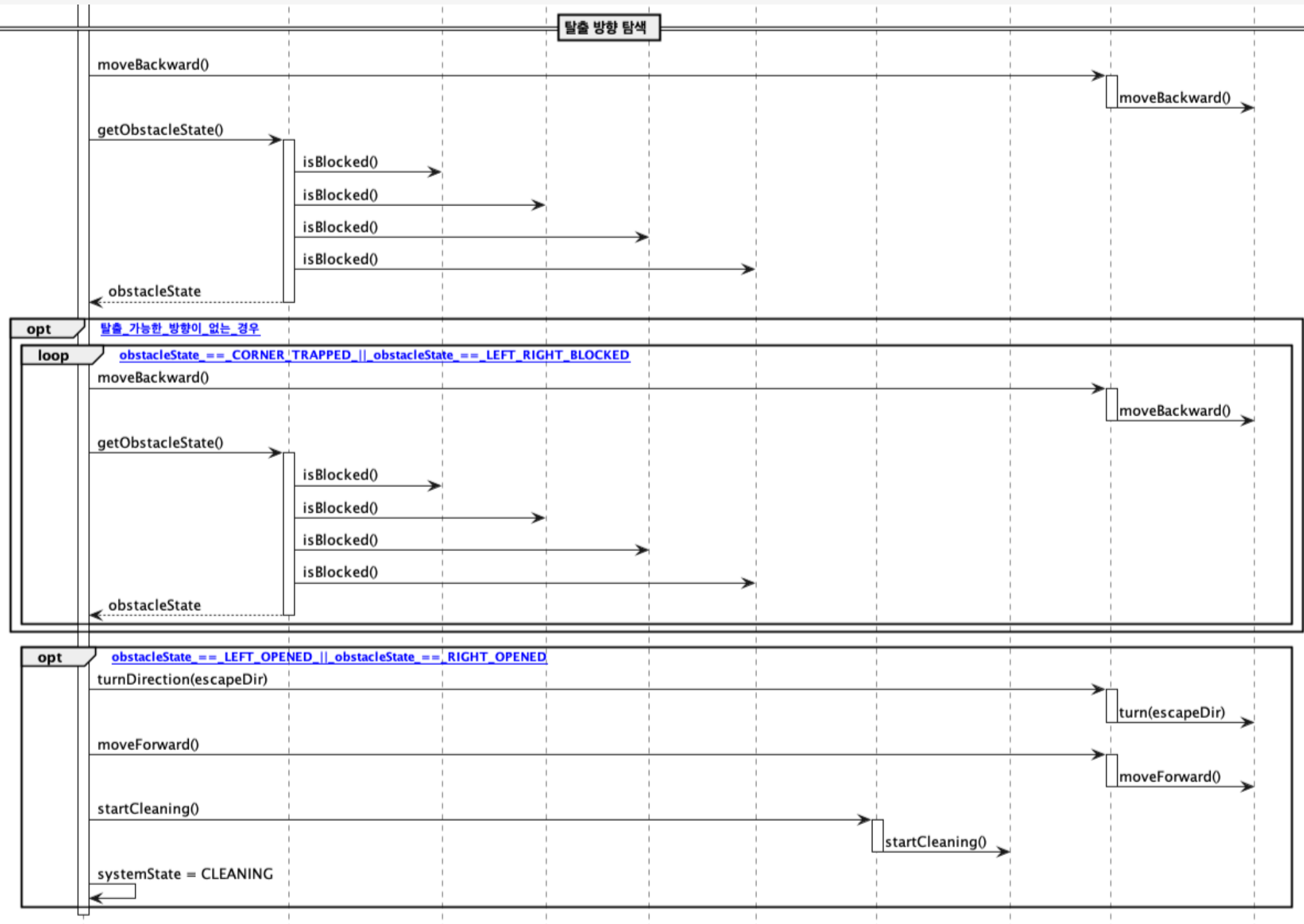
- UC-01에 의해 Orchestrator 내부에서 시작된 타이머가 센서 정보를 Poll
- 현재 시각이 강화 청소 종료 시각을 초과
→ 강화 청소 종료 요청
- 장애물이 없으며 먼지도 없는 경우
→ 일반 청소 모드
- 장애물 발견 여부에 따라
 - 전방 장애물 발견: UC-03
 - 전방 좌우 모두 막힌 경우: UC-04
 - 먼지가 감지된 경우: UC-05



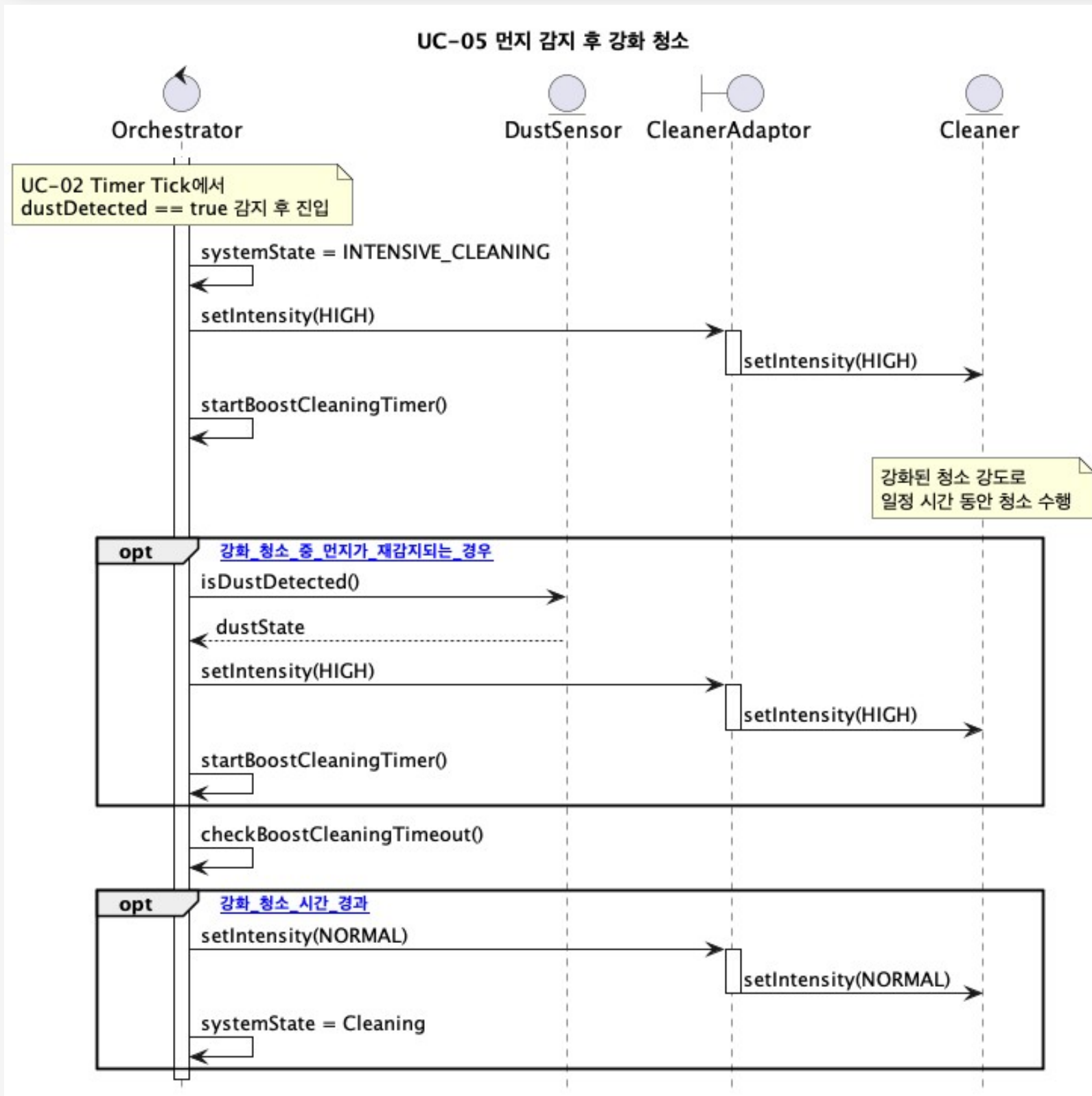
- UC-02 Timer Poll 과정에서 FRONT_BLOCKED 상황인 경우 실행
- Motor와 Cleaner를 정지
- 전방만 막힌 경우
- 좌회전 및 일반 청소 진행
- 전방 및 좌측이 막힌 경우
- 우회전 및 일반 청소 진행
- 전방 및 양측이 막힌 경우
- UC-04에서 처리



- Orchestrator Timer Tick에서 CORNER_TRAPPED인 경우 실행
- 시스템 상태를 REVERSING으로 변경
- 청소 정지 및 후진 실행
- loop: 후진 중 장애물을 만난 경우 (사방이 장애물인 경우) → 장애물이 사라질 때까지 센서 폴링

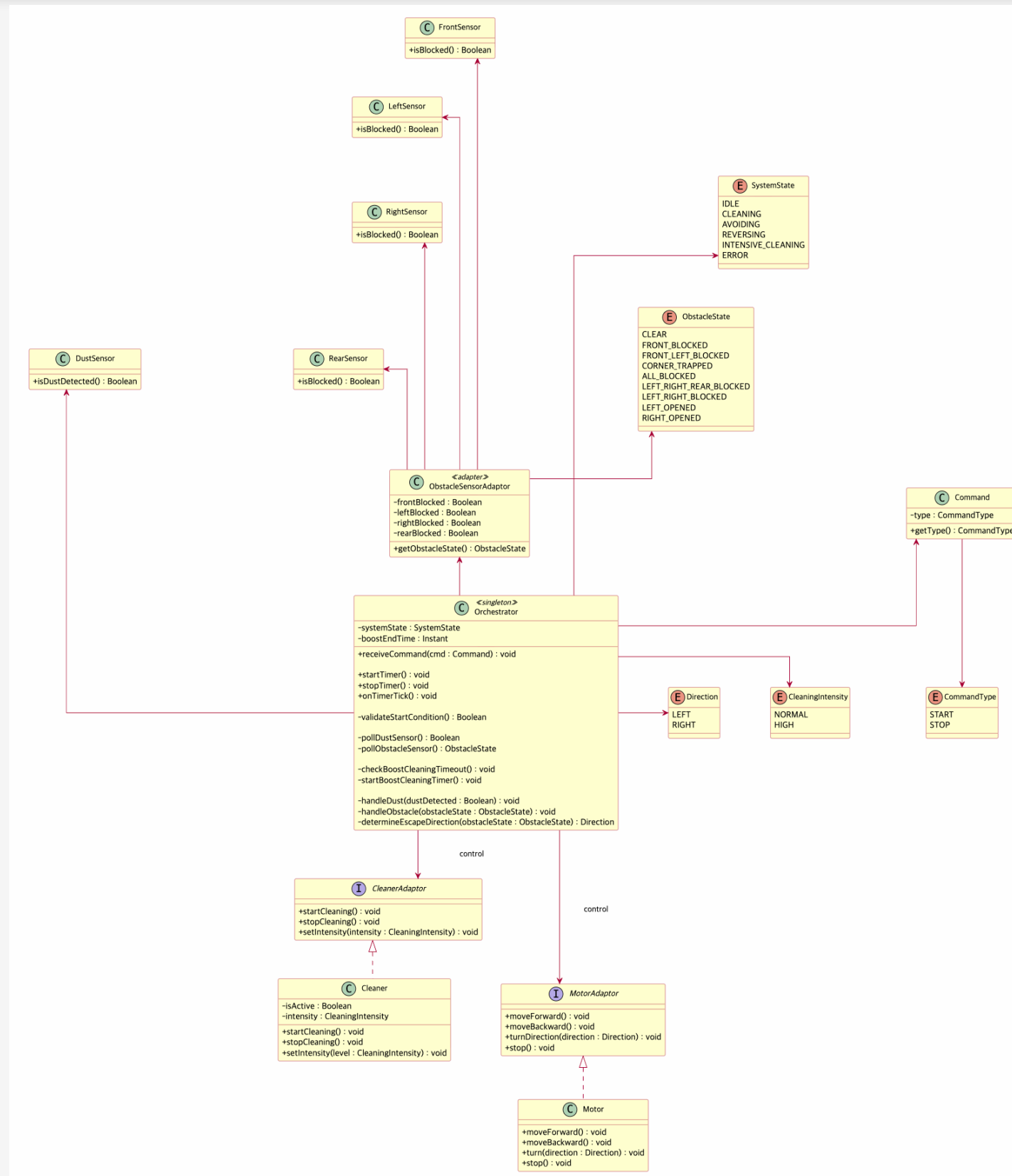


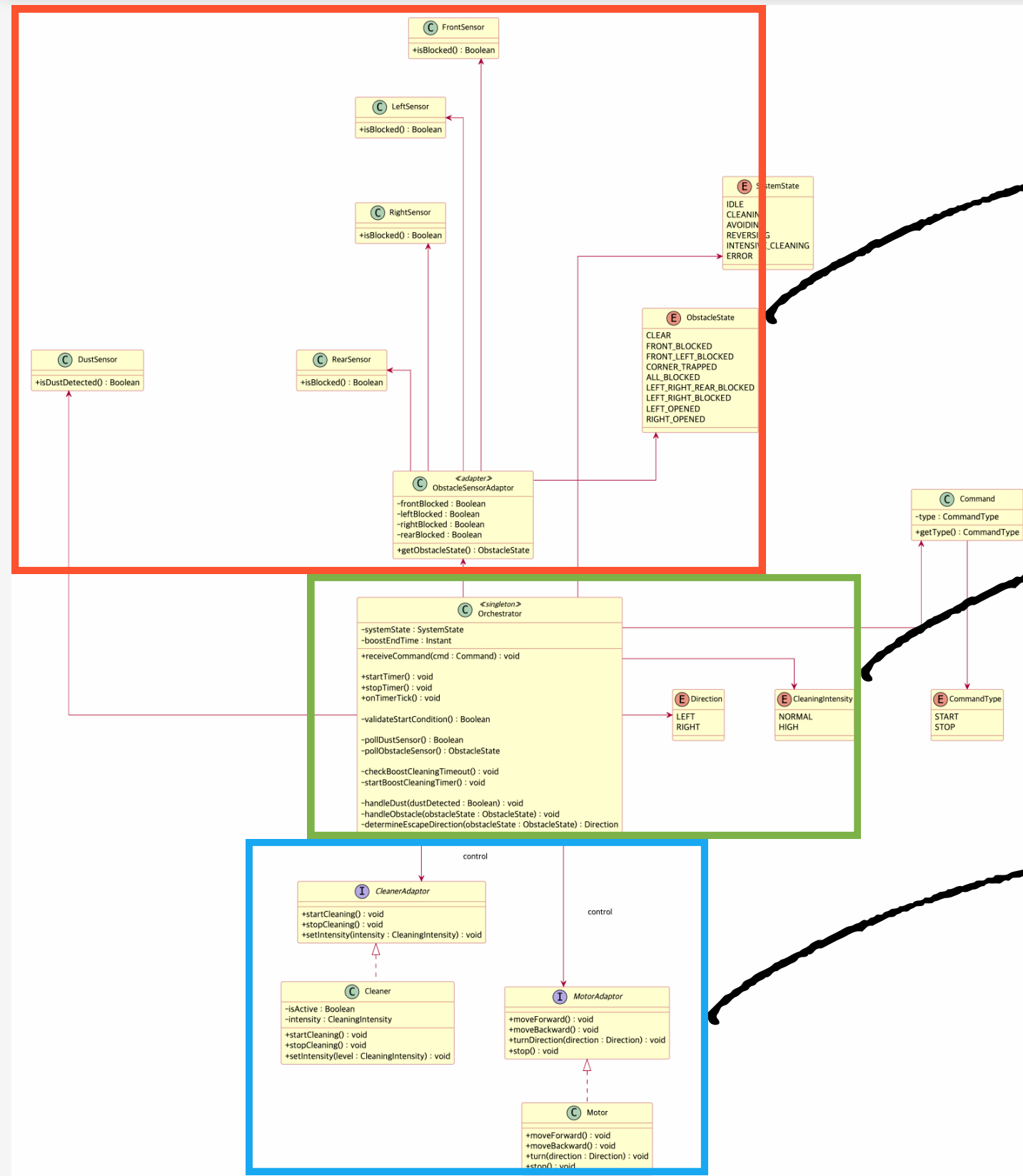
- 후진 후 탈출방향이 존재하는 경우
 - 좌측이 열려 있는 경우: 좌회전
 - 좌측이 막히고 우측이 열려 있는 경우: 우회전
- 좌우측 탈출 방향이 존재하지 않는 경우
 - loop: 계속 후진 및 센서 폴링



- Timer Tick에서 먼지를 발견한 경우 시작
- Cleaner 강도를 HIGH로 설정
- 강화 청소 종료 시각을 설정
 - 이후 Timer Tick에서 먼지가 재감지 되는 경우
 - 강화 청소 종료 시각을 갱신
- 현재 시각이 강화 청소 종료 시각을 초과
 - 일반 청소 모드로 변경

Class Diagram

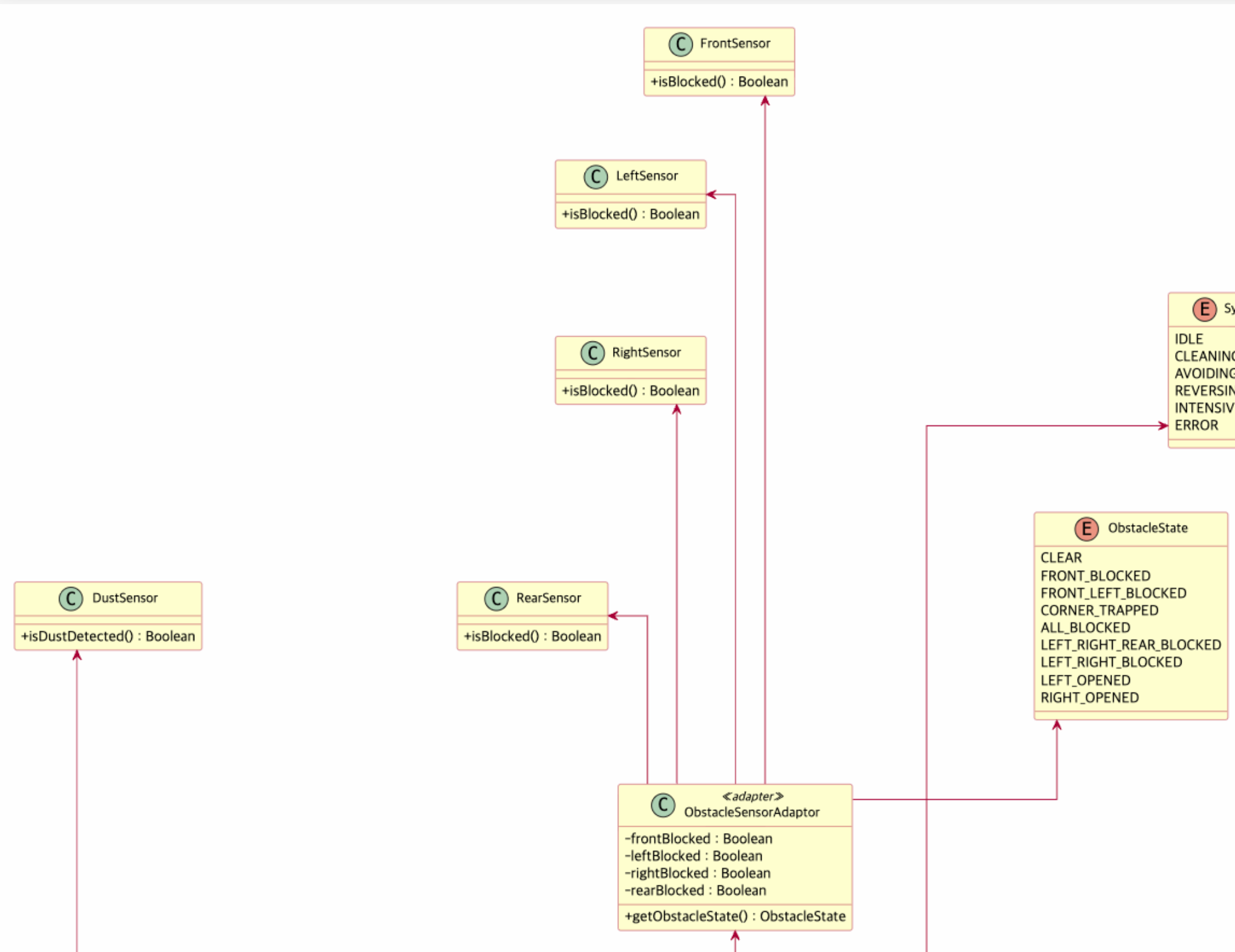




Input

Processing

Act

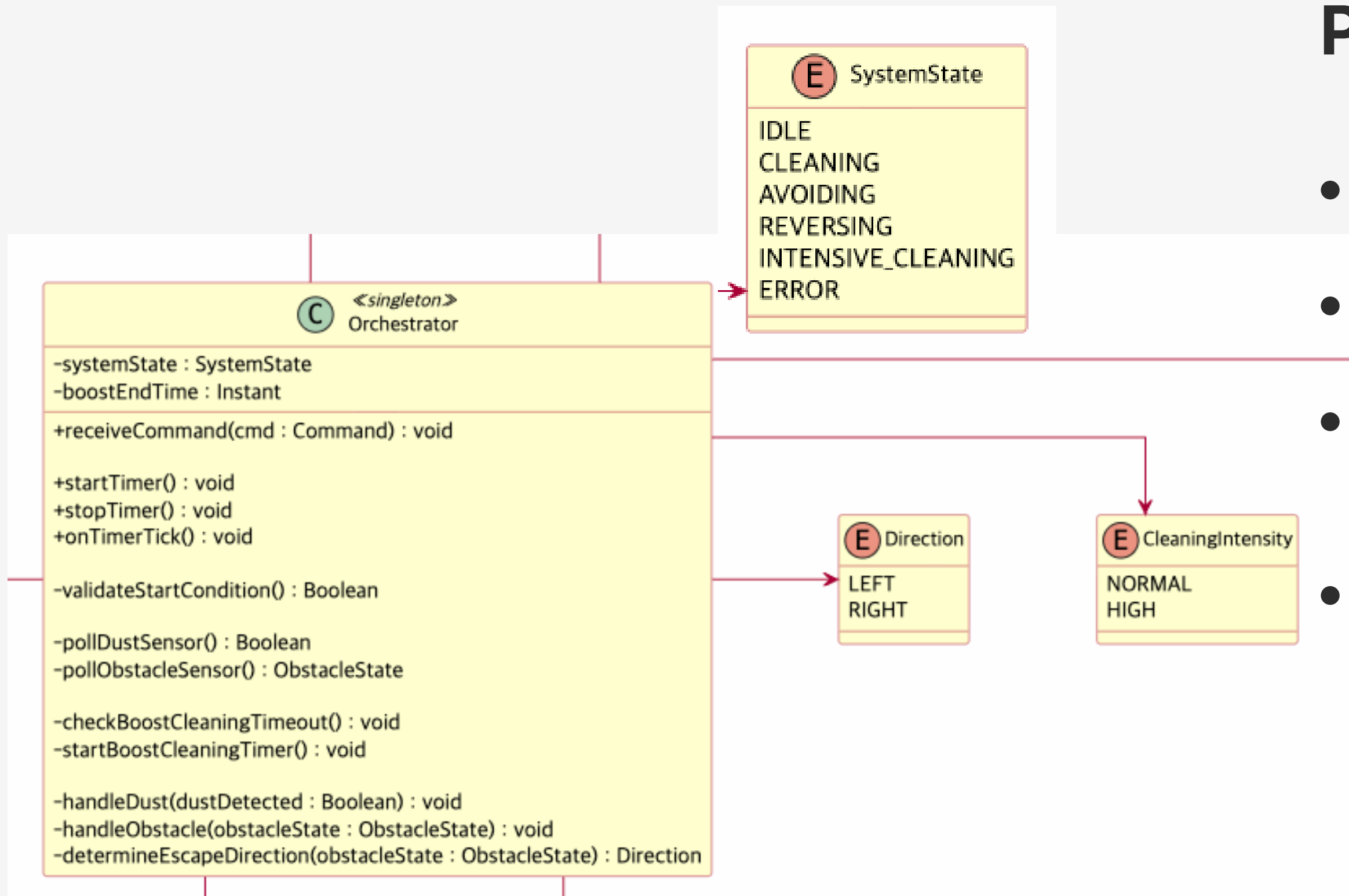


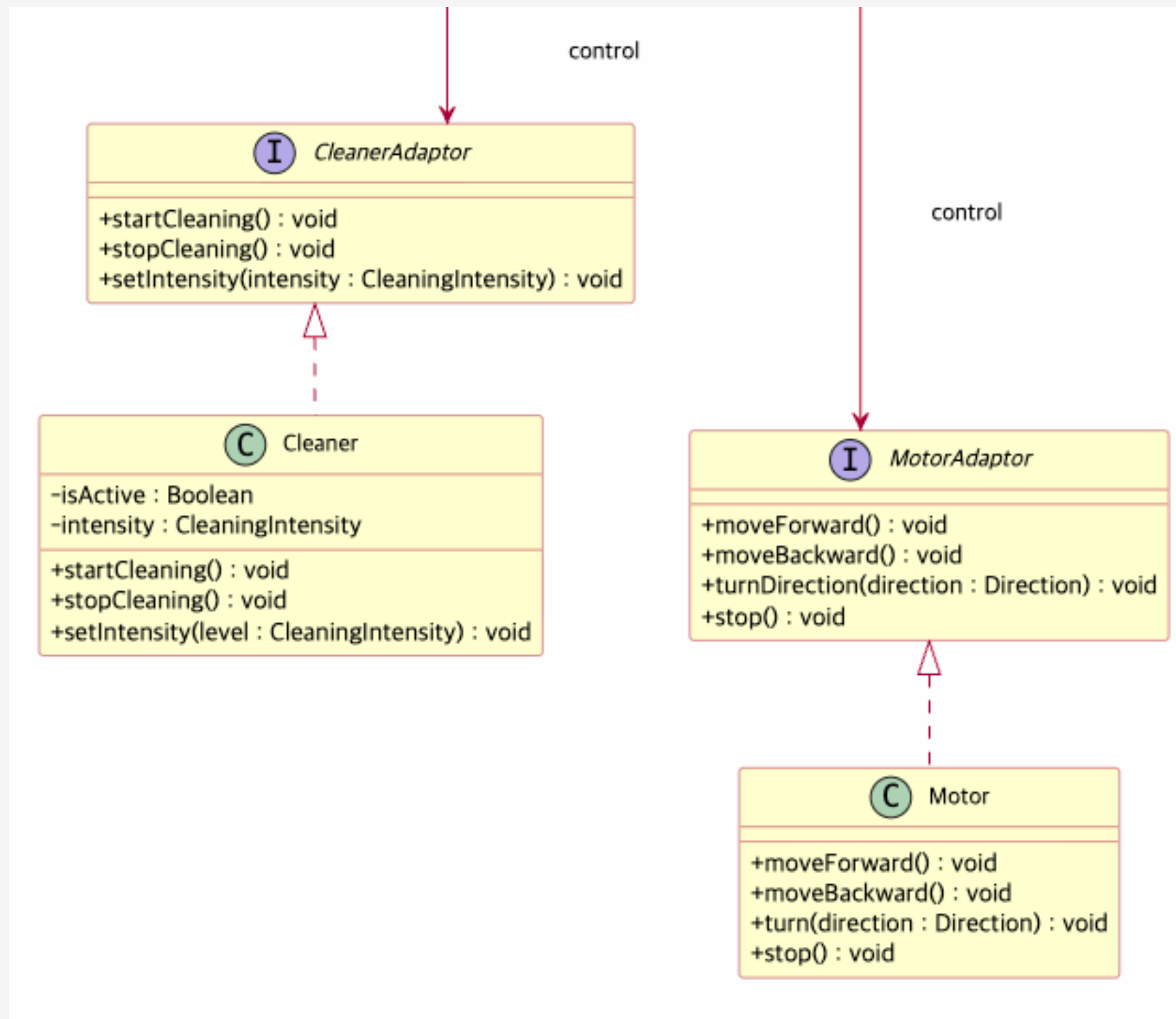
Input

- ObstacleSensorAdaptor 가 각 센서에서 받은 정보를 종합해 상태(ObstacleState) 를 판단

Processing

- Orchestartor 는 싱글톤으로 구현
- Orchestrator 는 내부 타이머를 사용해 센서 값을 폴링
- obstacleSensorAdaptor 와 dustSensor 에서 받은 상태를 바탕으로 전략을 생각
- 이후 모터나 클리너를 호출





Act

- Orchestrator 는 전략을 바탕으로 모터, 클리너에게 동작을 지시
- DI 패턴을 사용해 변경 및 테스트가 용이하도록 디자인

감사합니다