

RVC의 확장과 SA 과정 수정

철학과

201410191

김선홍

이해한 내용 정리

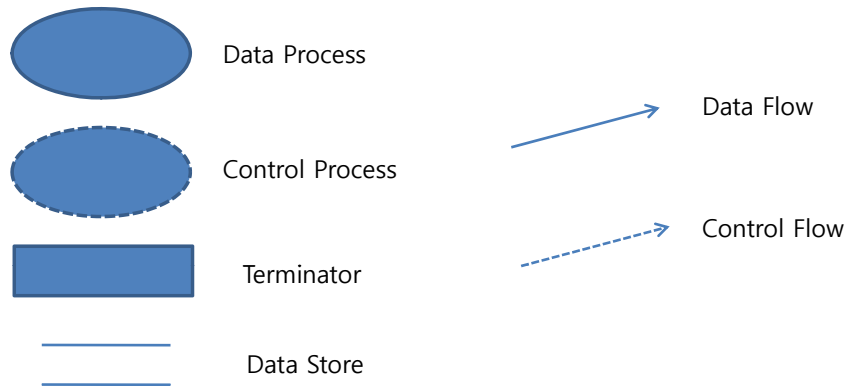
- 소프트웨어의 유형: 주문형 / 패키지형 / **임베디드**(하드웨어 장치에서 수행되는 소프트웨어로, 이 수업과 관련)
- SASD(Structured Analysis and Structured Design) : 구조적 분석 + 구조적 디자인, **절차지향적인 언어를 대상으로 함**
- SASD Process의 과정으로 모델링이 이루어짐

RVC – Statement of Purpose

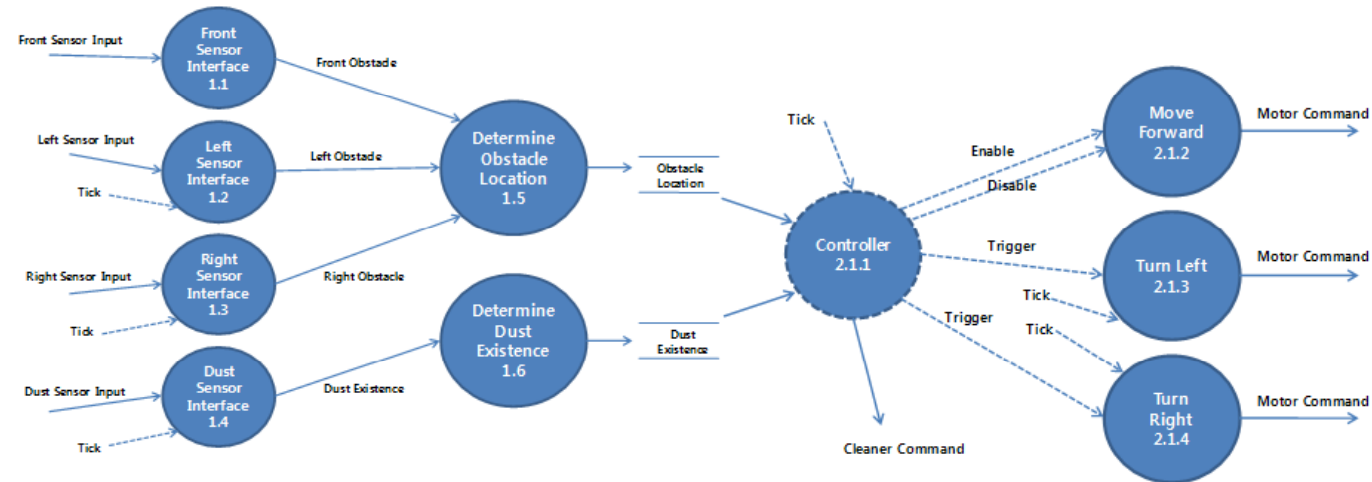
- Robot Vacuum Cleaner는 자동으로 집안표면을 청소하는 기계
- 청소하는 동안 곧장 앞으로 감
- 센서가 장애물을 발견하면 청소를 중지하고 옆으로 돌림
- 먼지를 감지하면 잠시 동안 청소 전원을 켜
- HardWare control에 대한 세부 설계 및 구현을 고려하지 않음
- **자동 청소 기능에만 초점을 맞춤**

Data Flow Diagram (DFD)

- Provides a means for functional decomposition.
- Composed of hierarchies(levels) of DFDs.
- Notation (A kind of CDFD)

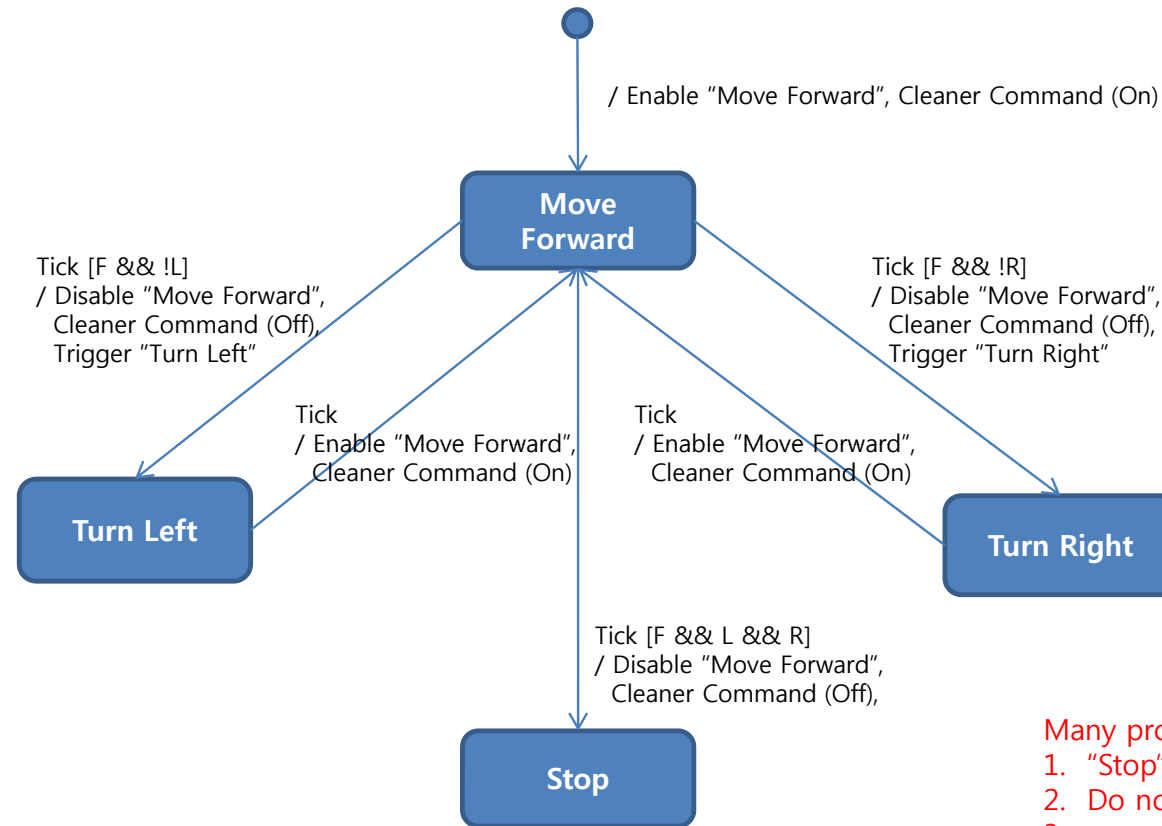


DFD – RVC Example



DFD Level 4 – RVC Example

State Transition Diagram for Controller 2.1.1



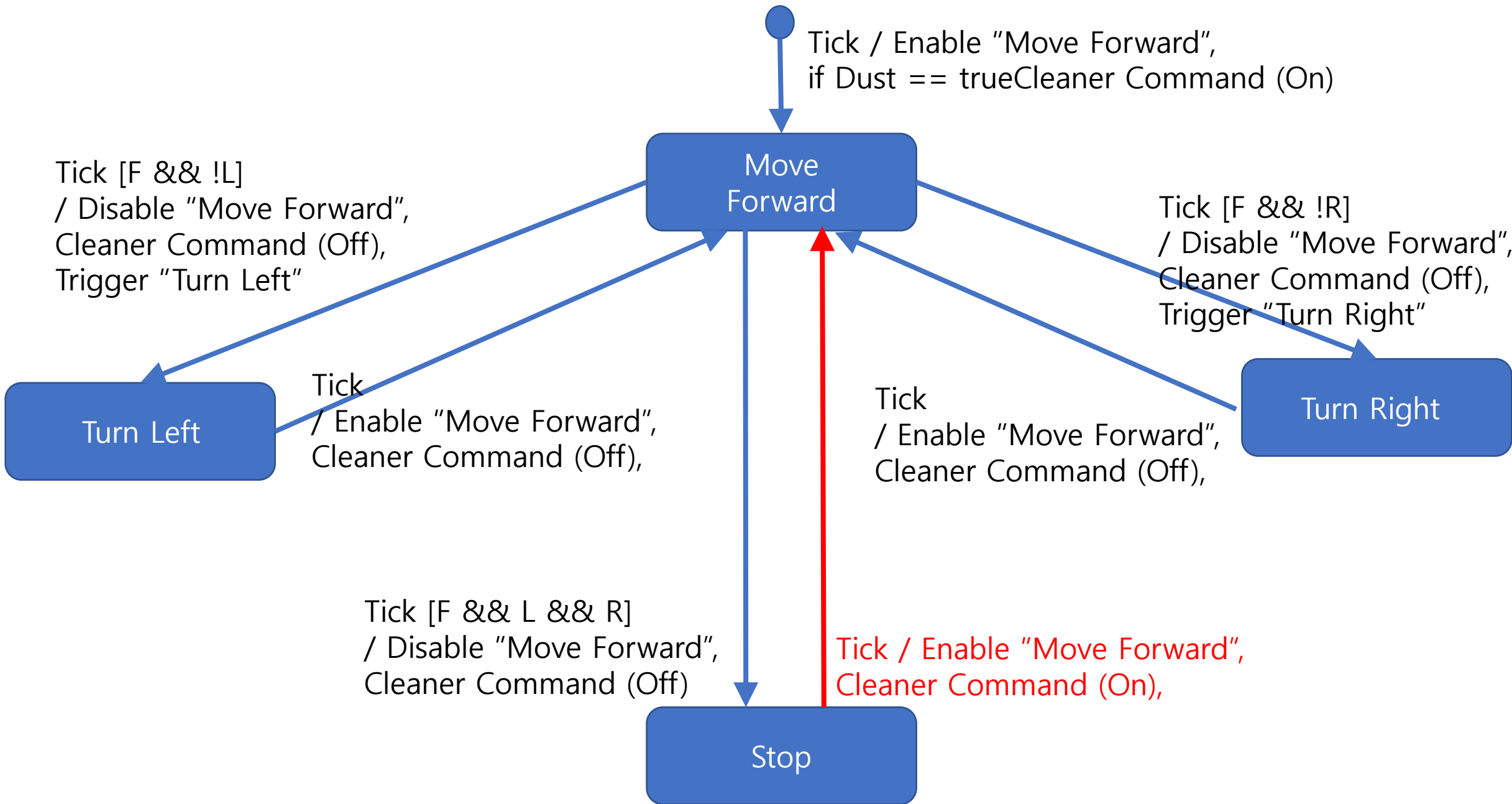
Many problems in this model:
1. "Stop" state
2. Do not consider "Dust"
3. ...

DFD Level 4 – RVC Example 의 문제점

- "STOP" state 문제: 멈춘 뒤, 다음 내용이 없음.
- Dust 문제: 먼지를 어떻게 할 것인지에 대한 내용이 없음
- 추가적인 요소 고려

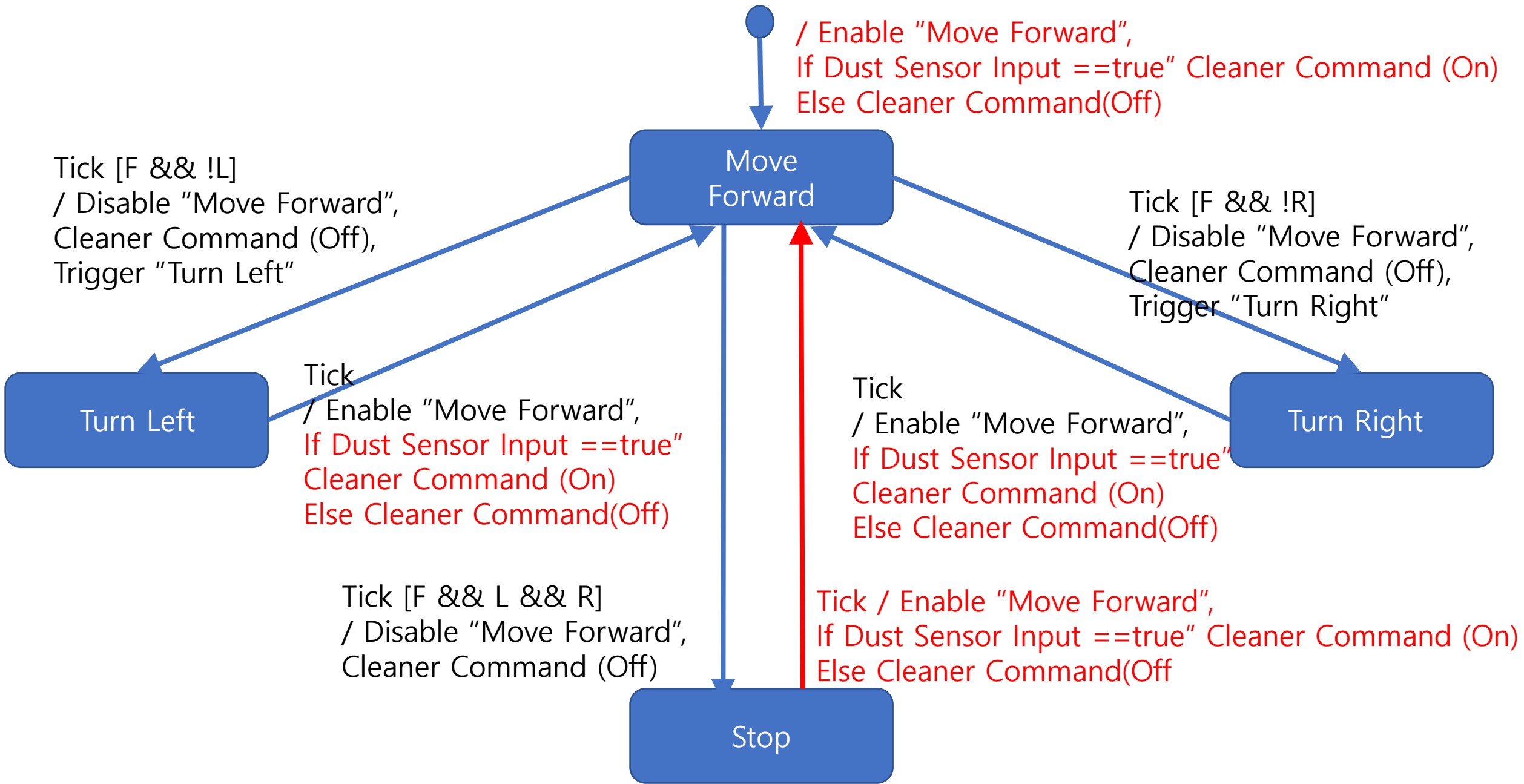
“STOP” State turn back

- 청소기는 네 가지(front,right,left,dust) 센서들로 장애물과 먼지를 감지합니다. 그리고 그러한 감지는 tick이라는 내부에 가지고 있는 시계를 통해서 주기적으로 이루어집니다.
- Stop하는 경우, Front, Left, Right로 갈 수가 없는 상황입니다. 따라서 다시 뒤로 가는 수밖에 없을 것입니다.



Consider "Dust"

- 청소기는 앞으로 갈 때 Command on이 됩니다.
- 컨트롤러는 장애물의 위치와, 먼지의 유무 데이터가 흐르는 곳입니다. 따라서 조건문을 걸어줘서, 먼지가 있으면 청소기를 on해주고, 먼지가 없다면, 청소기를 off하게 만들어줍니다. 또한 먼지의 감지 역시 내부 시계를 이용하여 주기적으로 이루어져야 합니다.



추가적인 보완점

- 먼지를 발견했을 경우, 얼마 동안 on할 것인지에 대한 고려가 필요합니다.
- 배터리에 대한 고려가 필요합니다.
- 회전 각도와 같은 이동에 대한 보완이 필요합니다. 만약 좌회전, 우회전이 90도로 이루어진다면, 사실 장애물이 있다고 가정했을 때, 모든 공간을 다 청소하는 게 불가능하기 때문입니다. 텅빈 공간이라면 벽 따라 돌기를 반복할 것입니다. 효율성을 고려하지 못하기 때문에, 자칫 같은 구간의 청소를 반복할 수 있습니다.
- 순서에 대한 고려가 필요합니다. 센서가 동시다발적으로 발생할 경우, 어디를 먼저 움직이게 될 것인지를 결정 해야합니다.

느낀점

- 천천히, 전체 과정을 읽어보면서 Statement of Purpose의 중요성을 크게 느꼈습니다.
- DFD level4의 다층적인 문제를 해결하기 위해선, DFD level을 거슬러 올라, 전체 구조에 대한 이해가 필요하단 걸 느꼈습니다.
- Data Dictionary와 설계의 기초가 되는 부분에 대한 단단한 이해가 필요한 것 같습니다.