

# 고급 소프트웨어 공학 : 스마트 살충제

# 목차

- 배경 및 목적
- 스마트 살충제 : Previous
- 스마트 살충제 : 유사사례/논문조사
- 스마트 살충제 : 객체 선정을 위한 시나리오
- 스마트 살충제 : 객체 선정
- 스마트 살충제 : 객체 관계 설정
- 스마트 살충제 : 검증 항목 선정
- 스마트 살충제 : 객체 상태 천이
- 향후 계획

# 배경 및 목적

- 모델링을 통해 발생 할 수 있는 요구사항을 도출 하고 그에 대한 검증을 해봄으로써 개발의 효율을 높이는데 목적이 있다.
- 순서는 다음과 같이 진행 될 예정으로 있다
  - ~~1차 요구사항 도출~~
  - ~~요구사항에 따라 필요한 객체 설정~~
  - 상태 머신 그리기 ← **진행 중**
  - 모델링 작성 및 CTL 작성
  - 2차 요구사항 도출

# 스마트 살충제 : Previous

- 1차 요구사항 도출
  - ~~곤충이 활발하게 움직이는 시기에만 작동 하도록 하는 온도 센서 필요~~
  - 곤충의 크기는 매우 작기 때문에 어느정도까지의 곤충을 방제 할 것인지에 대한 기준 필요(흰줄 숲모기와 빨간 집모기를 목표로...)
  - 일정 크기의 곤충이 근처에 있다는 것을 인식하기 위한 센서 필요
  - 노즐이 일정 기간 노후 되거나 하면 교체할 수 있는 시기에 대한 알림 필요
  - 모기를 유인하기 위한 이산화탄소 발생과 열 (300~400nm 의 파장이 짧은 자색계열)이 필요함

# 스마트 살충제 : 유사 사례/논문 조사

- 창의적 문제해결 이론(TRIZ)를 이용한 공기 정화 기능의 모기 유인 퇴치기 개발 2012.6 한국정밀공학회지
  - 본 내용은 제품이 자외선 램프/이온코팅/흡입팬/포집통 으로 구성 됨
  - 논문의 결과로 '블랙홀'이라는 상용제품을 도출(12만원) 하였으며 본 형태와 유사한 제품이 다수 출시 되어 있음
- 모기 퇴치용 램프 설계 및 구현 2001 대한전자공학회 종합학술 대회 논문집(하계)
  - 모기가 좋아하는 자외선 발생이 실험 결과를 통해 도출 되어 있음

# 스마트 살충제 : 유사 사례/논문 조사

- [LED로 모기 유인해 잡는 기술 개발](#) 2016.4.28 조선일보
  - '바이올레츠(violeds)' 라는 본 방법은 지카바이러스의 원인이 되는 이집트 숲모기와 말라리아를 옮기는 얼룩날개모기 등 해로운 모기가 좋아할 만한 파장의 자외선 파장의 패턴을 연구 하여 제품화 함
- [사물인터넷\(IoT\) 기반 모기지도 나온다.](#) 2017.4.27 서울시
  - 자동 모기 계측기의 사용에 대한 내용과 모기 발생 상황이 실시간으로 파악이 가능하다 라는 내용이 요지
  - 2010년에 유사한 기사로 구로구 에서 적외선 감지 센서로 모기가 들어올 때 마다 숫자를 세어 방역본부에 알리고 방역을 할 수 있도록 하는 부분을 언급한 유사기사가 있음

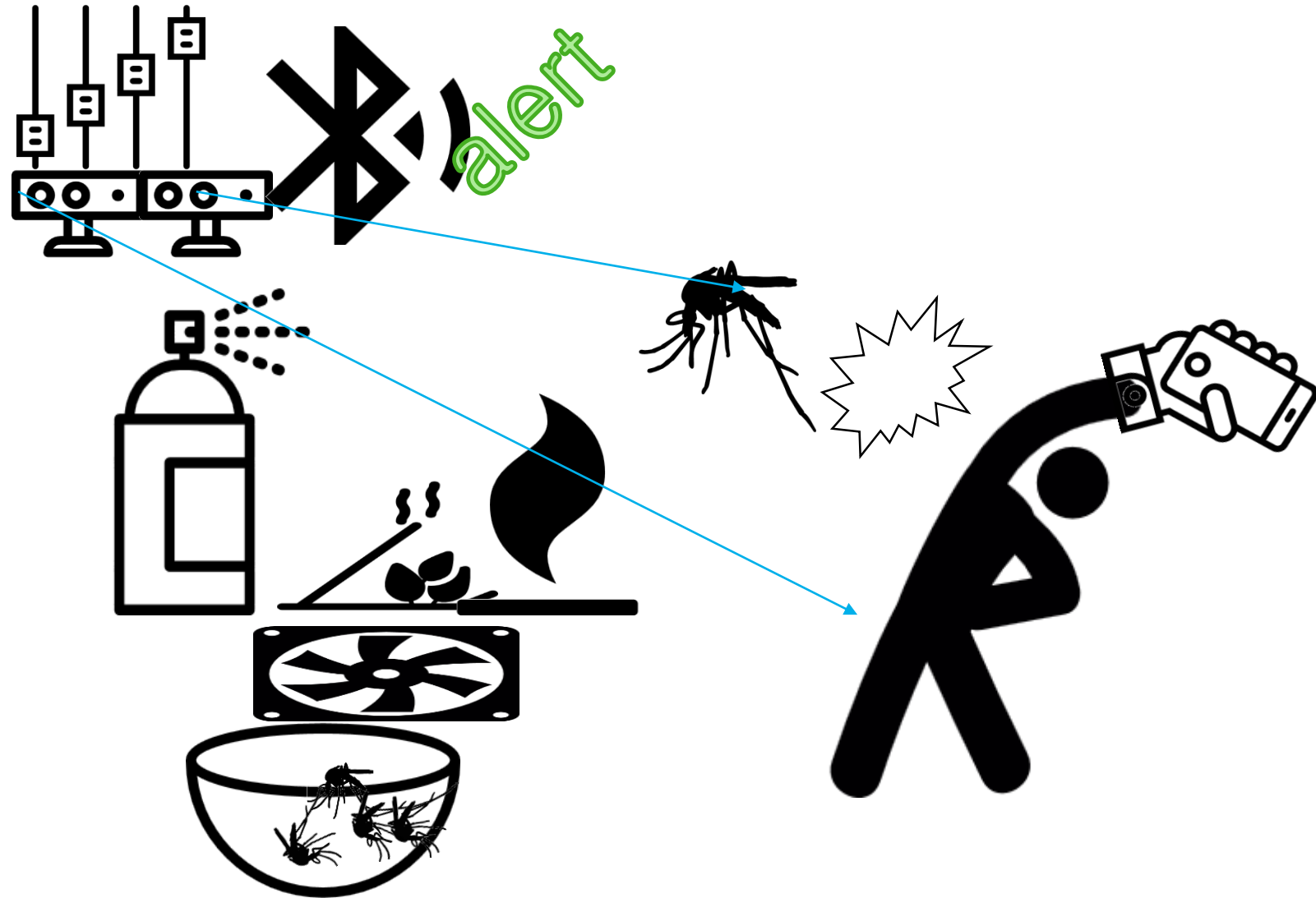
# 스마트 살충제 : 객체 선정을 위한 시나리오

- 모기 포집 순서

- 유인 → 모기인식센서 등으로 타깃 발견 → 살충 제품 살포 → 팬 작동 하여 포집

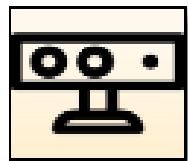
- 사용자의 역할

- 수동으로 중단
- 사람센서 앞에 등장
- 스마트폰으로 알림을 받은 후 조치

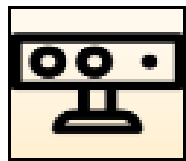


# 스마트 살충제 : 객체 선정

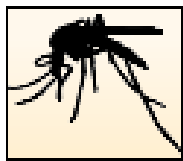
- 유인 : 이산화탄소 발생기, 열판(자외선), 모기 감지 센서
- 살충 : 살충제 pusher, 빨아들이는 방향의 팬, 포집기
- 편의 : 사람인식 센서(하부센서), 알림용 블루투스 LE 모듈(스마트폰으로 제제를 뿌린 횟수에 따라 교체 시기 알림, 이산화탄소발생기 타임 스탬프 기준으로 이산화탄소 발생기 교체 시기 알림)
- 모델 체크용 : target(모기), friend(사용자)



Target Sensor



Friend Sensor



Target



bowl



spray



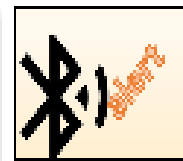
fan



friend



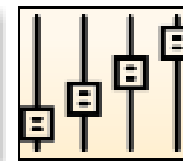
hotplate



bluetooth



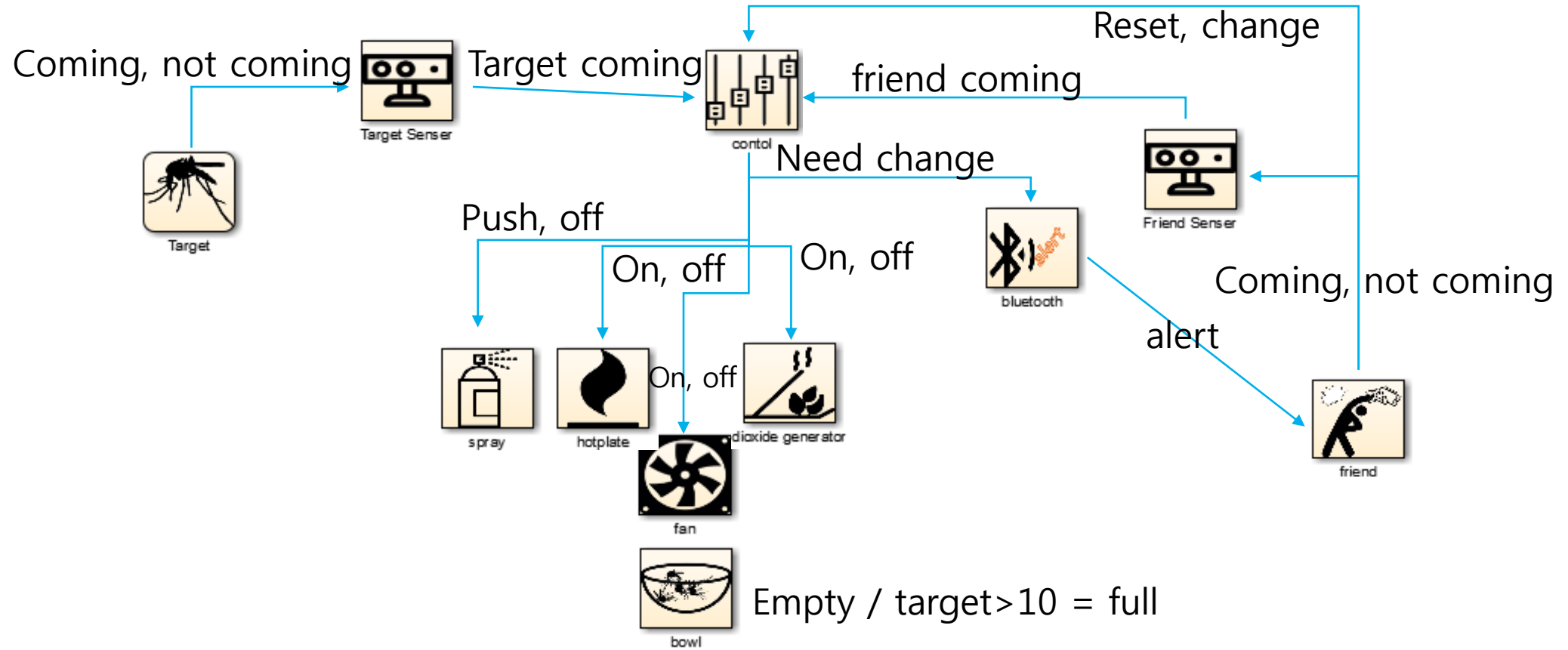
dioxide generator



control



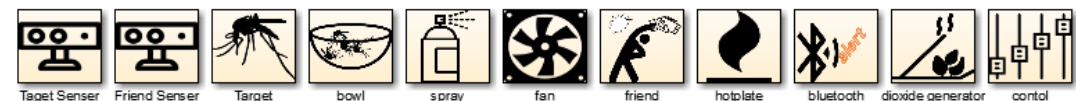
# 스마트 살충제 : 객체 관계 설정



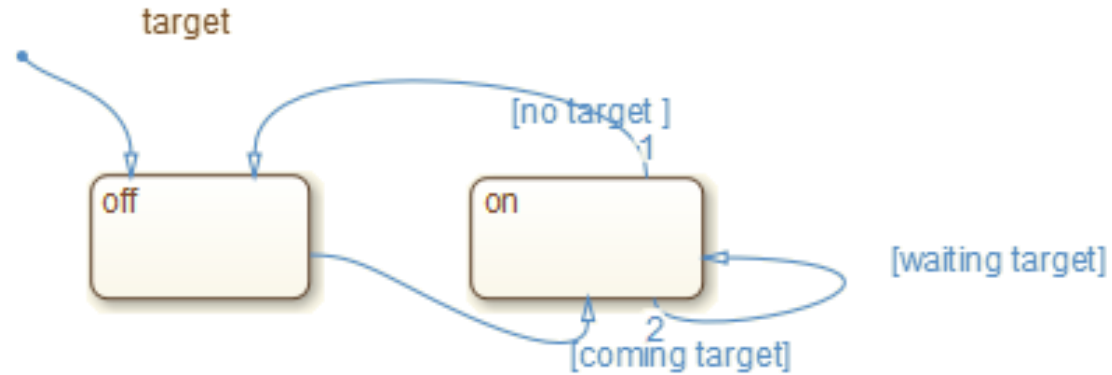
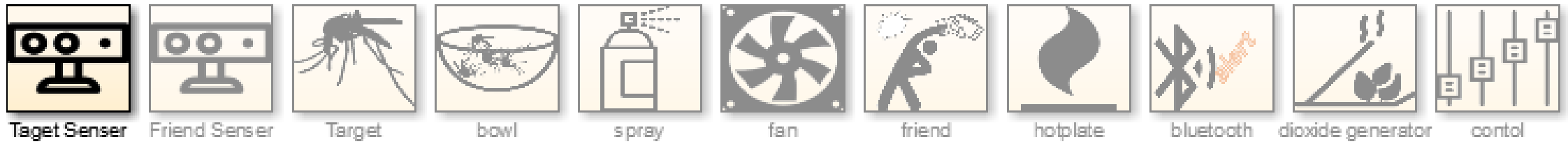
가상 변수  
 Alert : Spray capacity > 30  
 dioxide > 30

# 스마트 살충제 : 검증 항목 선정

- 사용자는 언제든지 수동으로 기기를 중단 시킬 수 있어야 한다.
  - 기기의 중단은 이산화탄소발생기, 열판, 살충제 pusher 를 포함하여 모두 중단
- 살충제는 센서 감지가 되지 않을 때 뿌려지면 안된다.
  - 센서 감지 전까지는 대상이 없으므로 계속 동작 시킨다.
- 이산화 탄소 발생기만 on 상태이거나 열판만 on 상태 일수 없다.
- 살충제 pusher는 사용자가 인식 되면 뿌려지면 안된다.
- 기기를 설치한 후 n번 이상의 살충제 pusher 가 눌러진 상태면 휴대폰으로 교체 알림을 전송해야 한다.
  - 블루투스 연결이 on 된 상태여야 함

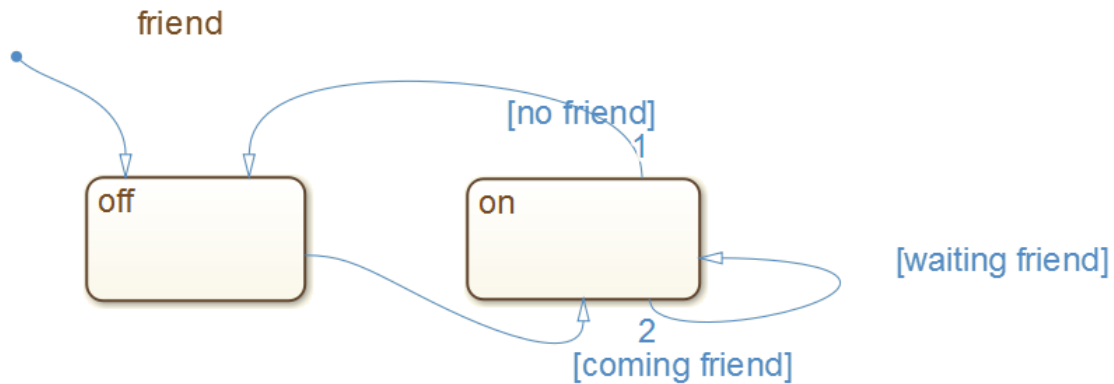
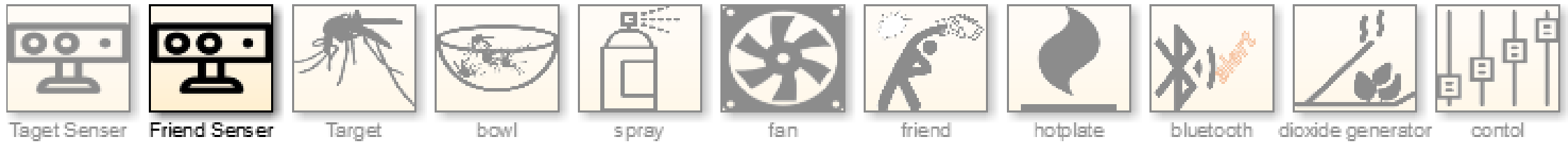


# 스마트 살충제 : 객체 상태 천이



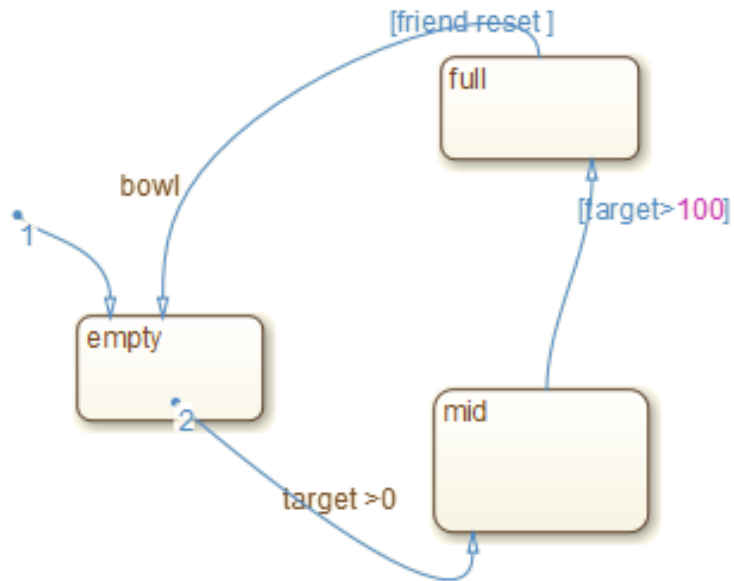
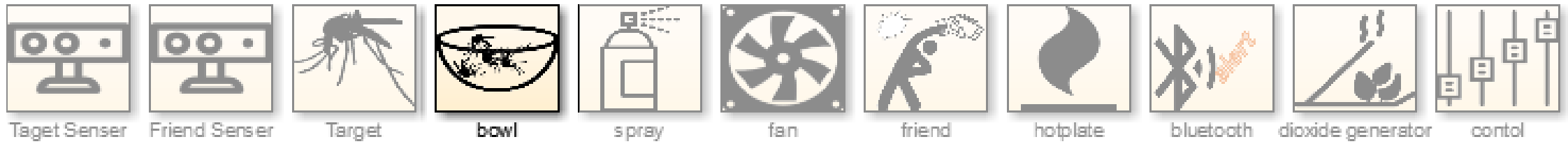
```
MODULE Target_Sensor(target)
  VAR
    status : { off, on };
  ASSIGN
    init(status) := off;
    next(status) :=
      case
        target = coming           : on;
        target = wating           : on;
        target = not_coming       : off;
      esac;
```

# 스마트 살충제 : 객체 상태 천이

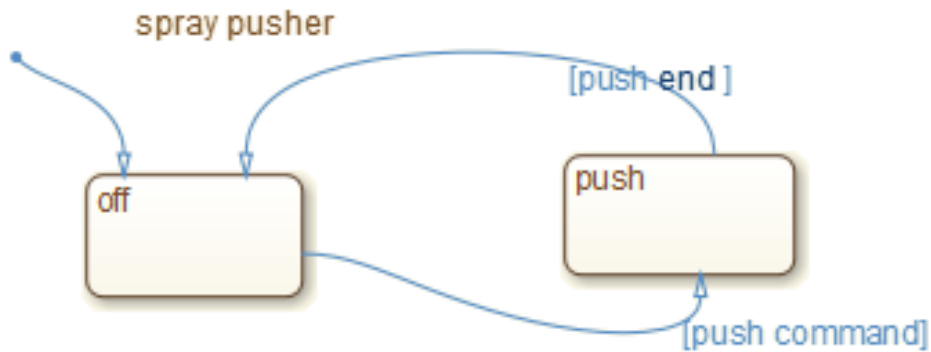
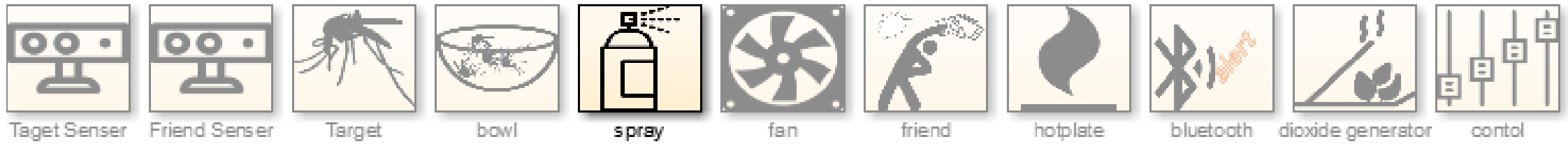


```
MODULE Friend_Sensor(friend)
  VAR
    status : { off, on };
  ASSIGN
    init(status) := off;
    next(status) :=
      case
        friend = coming      : on;
        friend = wating      : on;
        friend = not_coming  : off;
      esac;
```

# 스마트 살충제 : 객체 상태 천이

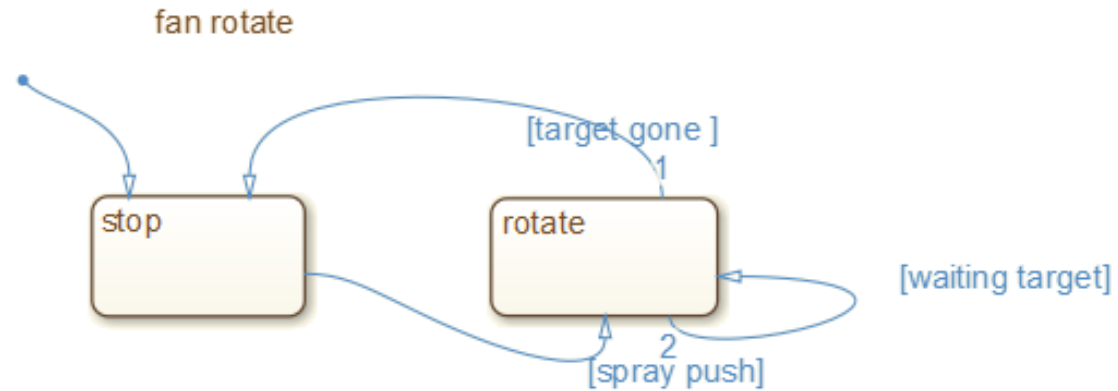
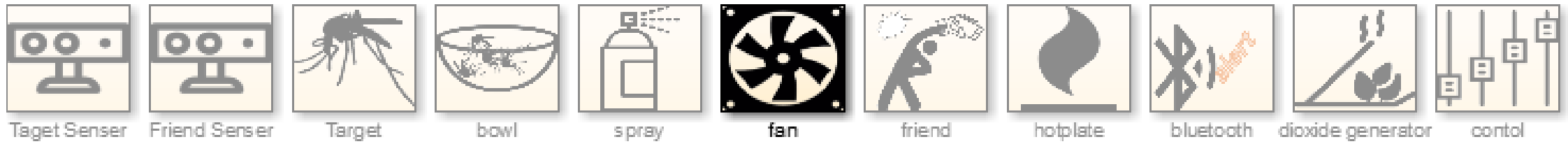


# 스마트 살충제 : 객체 상태 천이



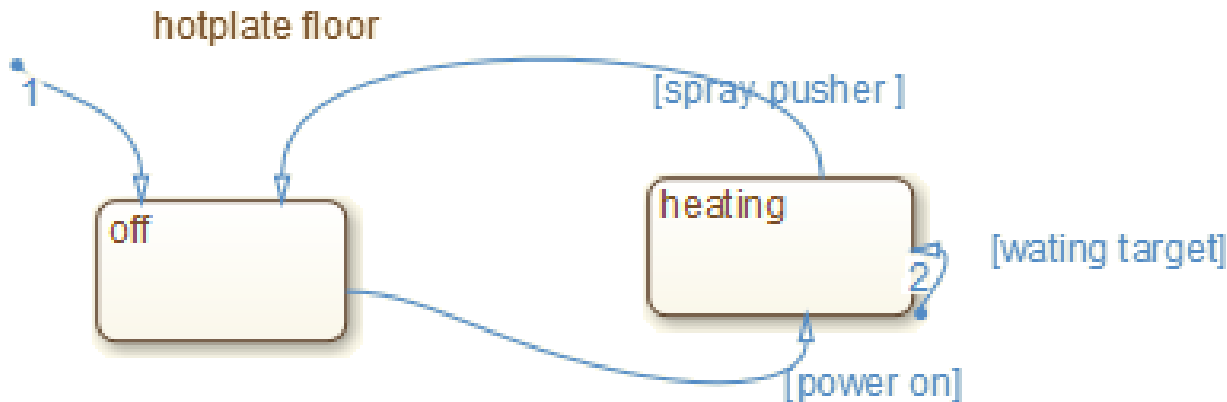
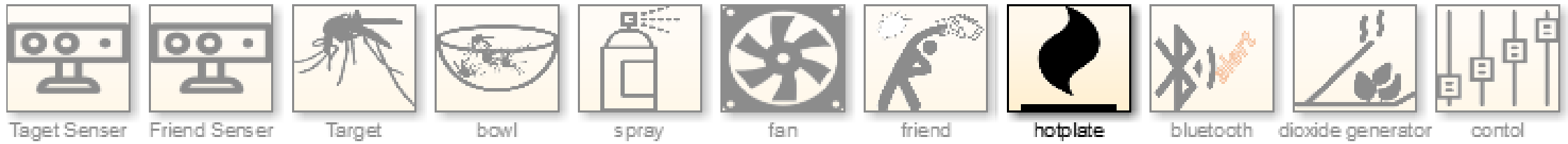
```
MODULE Spray(con_request)
  VAR
    Spray_pusher : { off, on };
  ASSIGN
    init(status) := off;
    next(status) :=
      case
        con_request = spray_push      : on;
        con_request = target_kill     : off;
      esac;
```

# 스마트 살충제 : 객체 상태 천이



```
MODULE Fan(request)
  VAR
    Fan_rotate : { stop, rotate };
  ASSIGN
    init(status) := stop;
    next(status) :=
      case
        con_request = spray_push      : rotate;
        con_request = target_kill    : stop;
      esac;
```

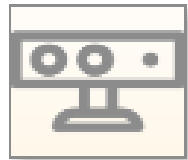
# 스마트 살충제 : 객체 상태 천이



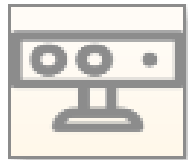
```
MODULE HotPlate(request)
  VAR
    HotPlate_floor : { off, heating };
  ASSIGN
    init(status) := off;
    next(status) :=
      case
        con_request = wating_target      : on;
        con_request = need_reset        : off;
        con_request = need_change       : off;
      esac;
```



# 스마트 살충제 : 객체 상태 천이



Taget Sensor



Friend Sensor



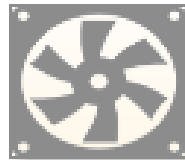
Target



bowl



spray



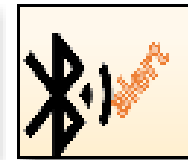
fan



friend



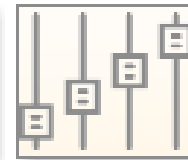
hotplate



bluetooth



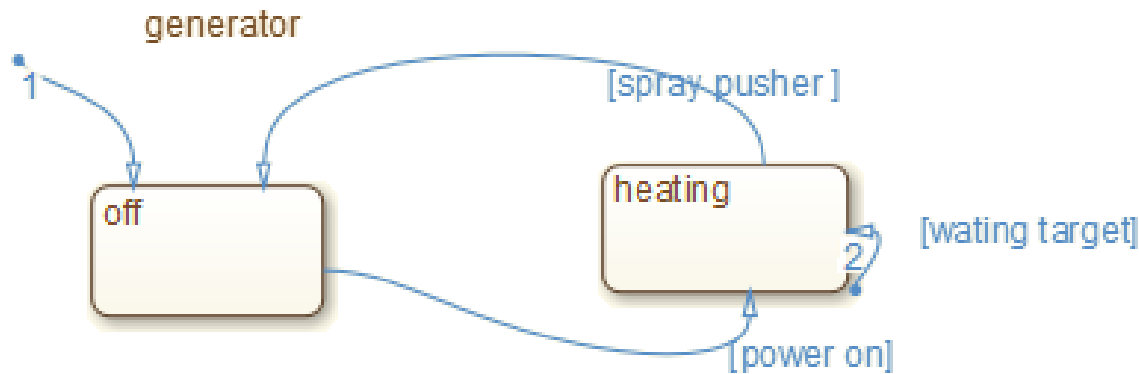
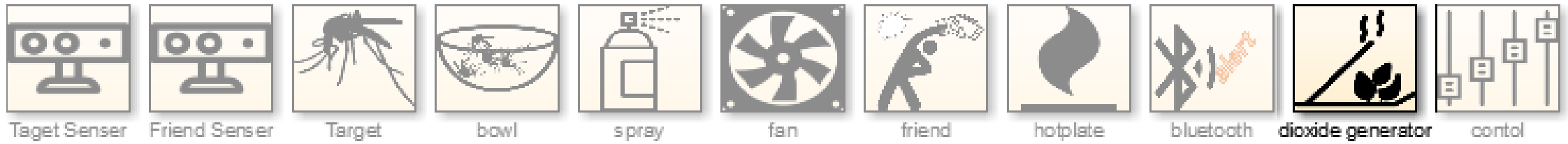
dioxide generator



control

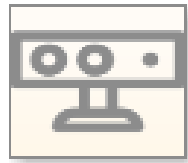
작업중

# 스마트 살충제 : 객체 상태 천이

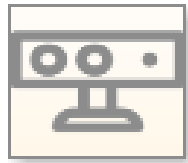


```
MODULE Dioxide_Generator(request)
  VAR
    generator : { off, heating };
  ASSIGN
    init(status) := off;
    next(status) :=
      case
        con_request = wating_target      : on;
        con_request = need_reset        : off;
        con_request = need_change       : off;
      esac;
```

# 스마트 살충제 : 객체 상태 천이



Target Sensor



Friend Sensor



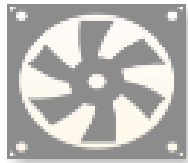
Target



bowl



spray



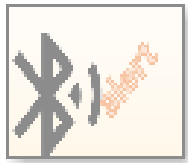
fan



friend



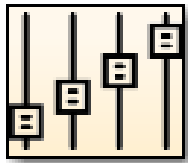
hotplate



bluetooth



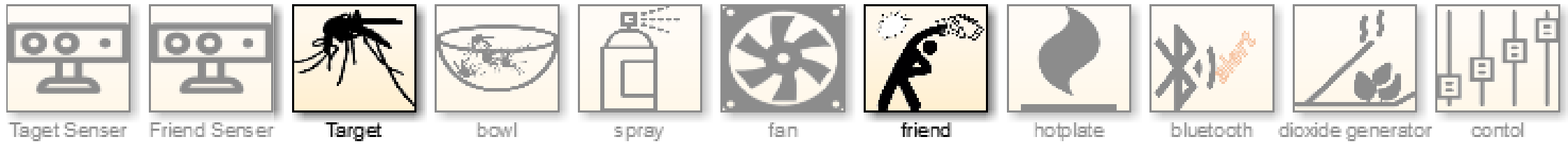
dioxide generator



control

작업중

# 스마트 살충제 : 객체 상태 천이



```
target: {coming, not_coming, wating};  
friend: {coming, not_coming, wating, need_reset, need_change};
```

# 향후 계획

- SMV 로 state 기준 객체와 검증 내용 작성 하여 추가적으로 발생하는 요구사항 도출 및 객체 수정 예정
- 진행순서
  - ~~1차 요구사항 도출~~
  - ~~요구사항에 따라 필요한 객체 설정~~
  - 상태 머신 그리기 ← **진행 중(알림용 Bluetooth, controller)**
  - 모델링 작성 및 CTL 작성
  - 2차 요구사항 도출