An Introduction to SoarML

Design

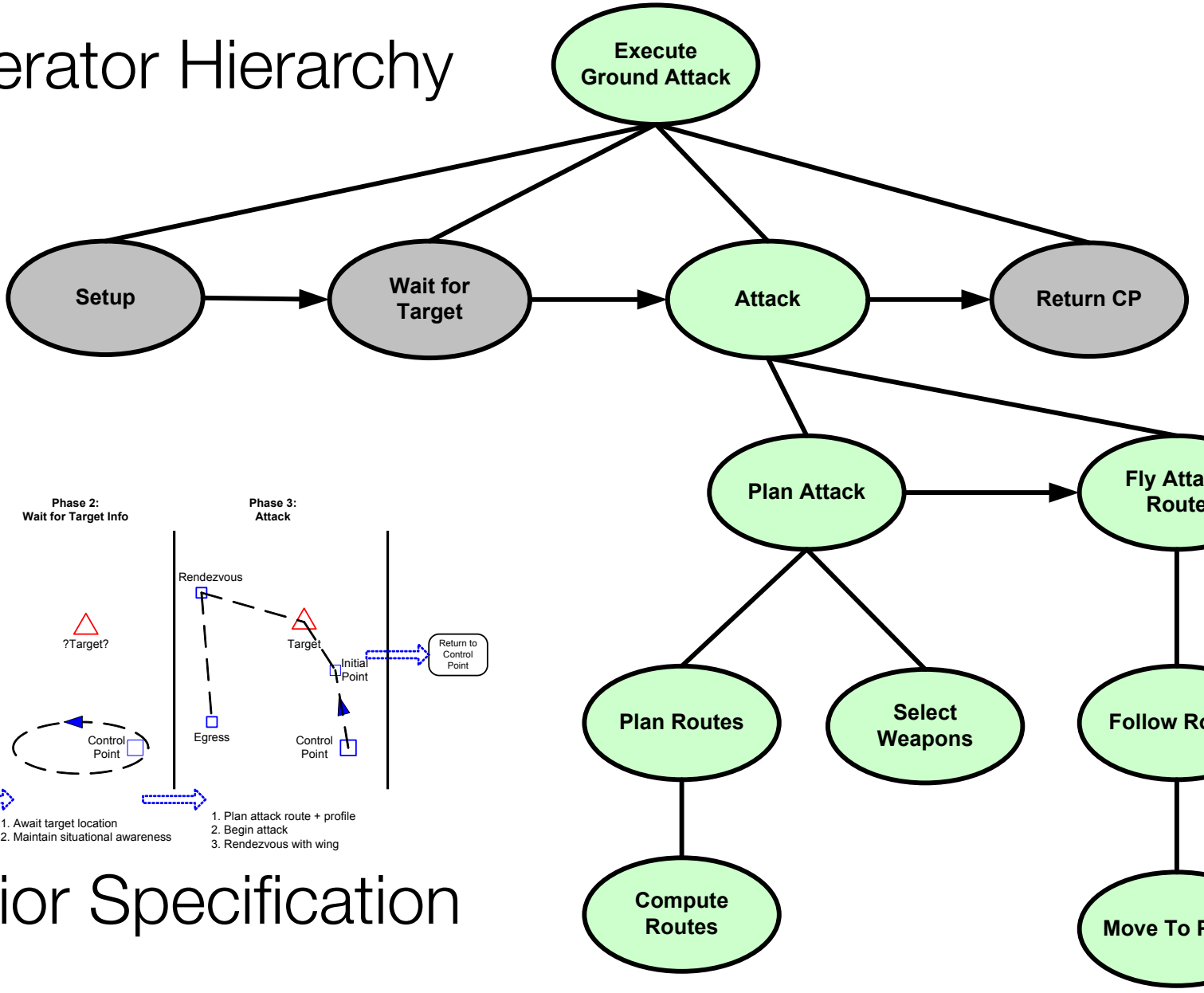
An abstraction that defines the constraints, form, and function of your software

Allows manipulation and analysis
Directs implementation (what and how)
Facilitates human communication and review
Suggests ways to share and reuse software

At Soartech you sometimes here: “Listen to the architecture”
This means let the design of Soar guide your implementation

Design and Soar

Operator Hierarchy

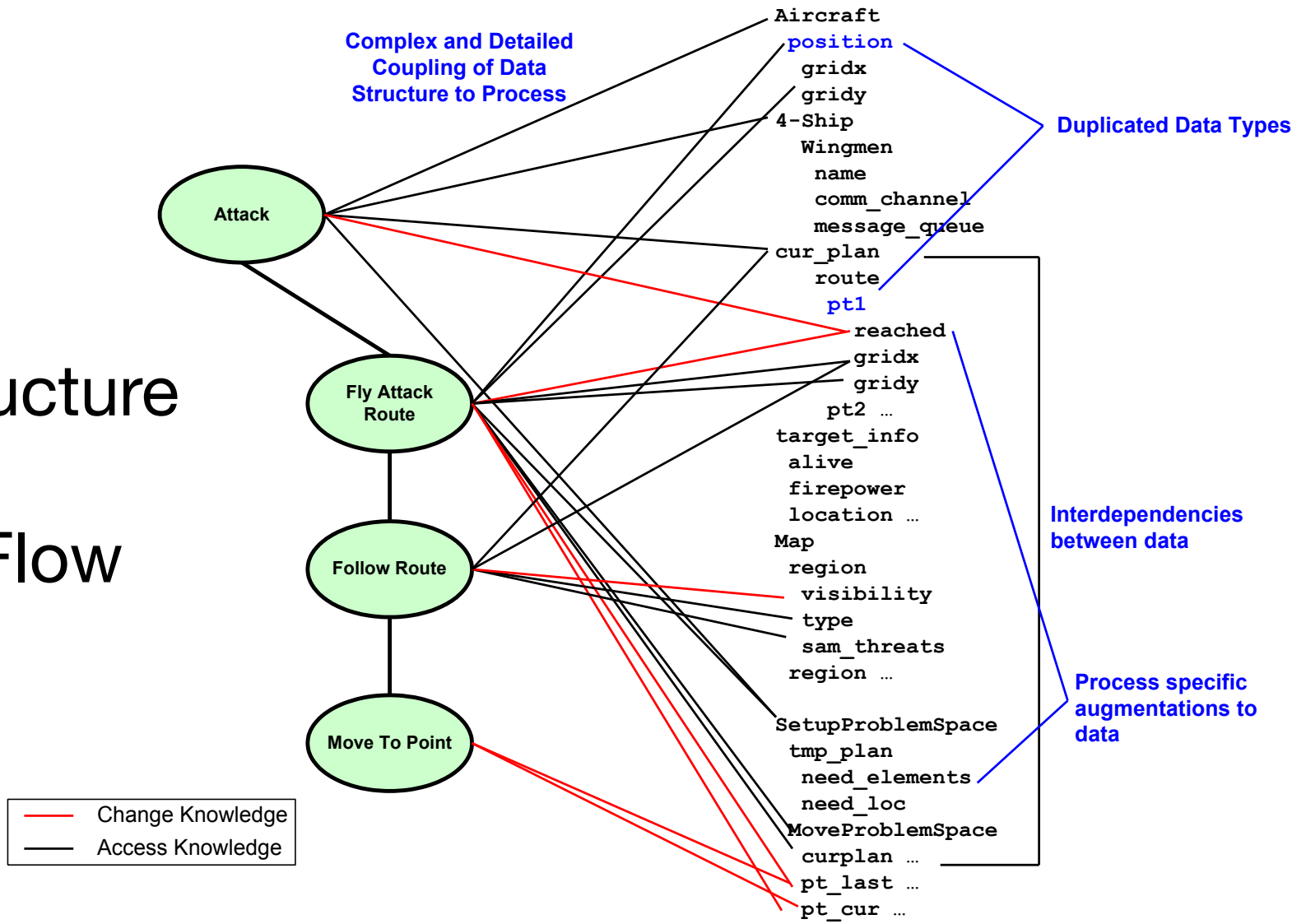


Behavior Specification

What is Missing?

Data Structure

Control Flow



HLSR work drove SoarML development

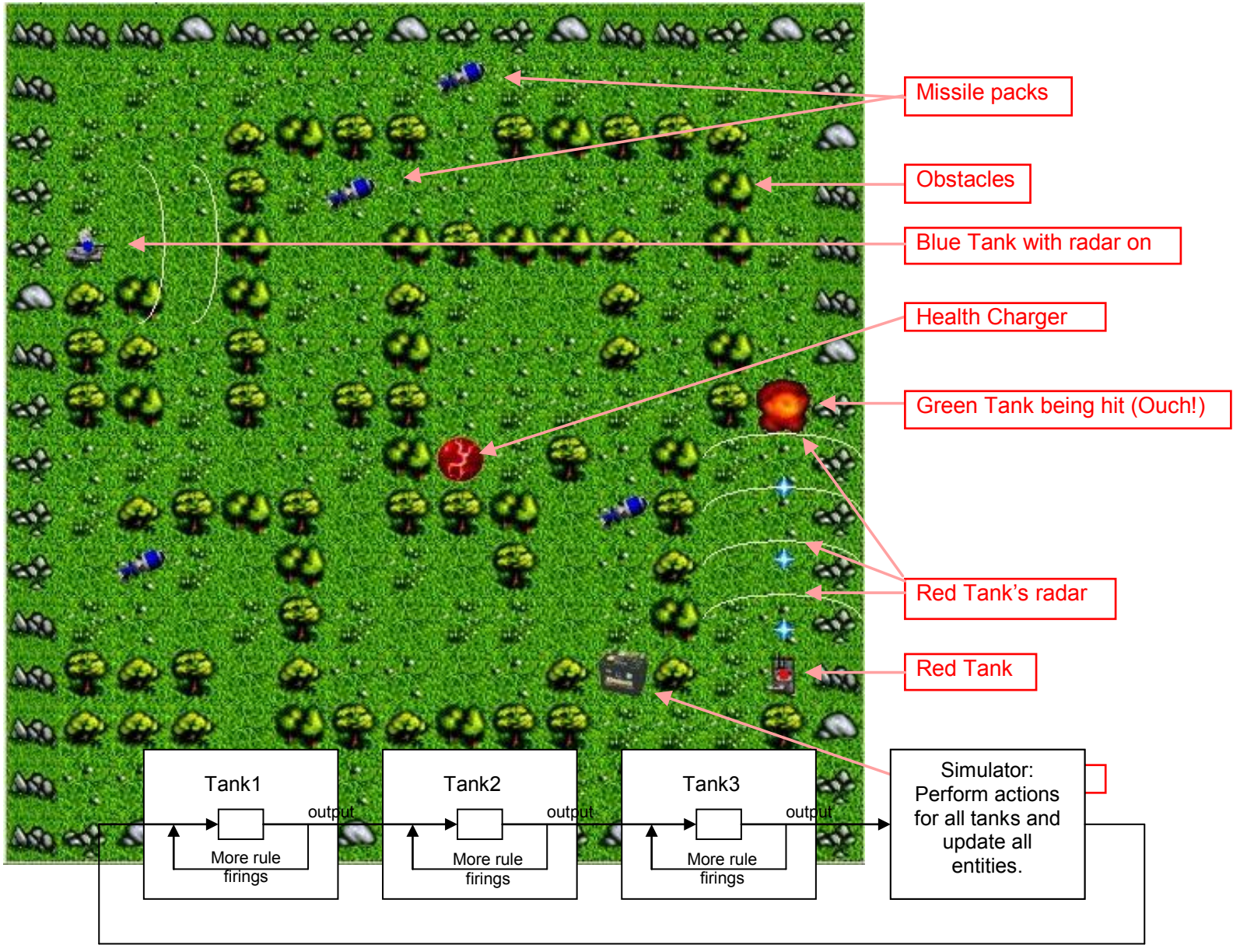
Elements of Soar Design

Working Memory Structure
(borrow from UML ojbects)

Goals
(borrow from agent design languages)

Processes
(blends goals and memory with operators)

Tank Environment



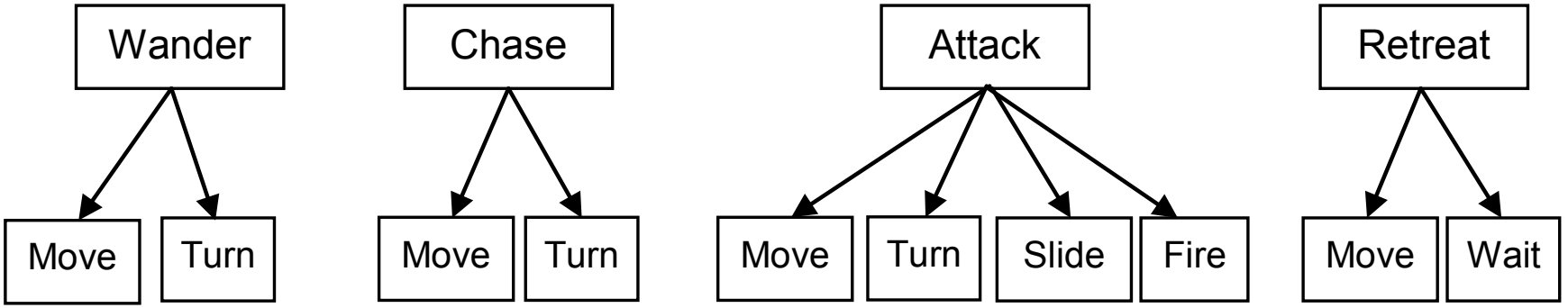
Overview of IO

```
^io
   ^input-link
      ^blocked
         ^backward yes/no
         ^forward yes/no
         ^left yes/no
         ^right yes/no
      ^incoming
         ^backward yes/no
         ^forward yes/no
         ^left yes/no
         ^right yes/no
      ^radar
         ^energy
            ^distance 0-13
            ^position left/center/right
         ^health
            ^distance 0-13
            ^position left/center/right
         ^missiles
            ^distance 0-13
            ^position left/center/right
         ^obstacle
            ^distance 0-13
            ^position left/center/right
         ^open
            ^distance 0-13
            ^position left/center/right
         ^tank
            ^distance 0-13
            ^position left/center/right
```

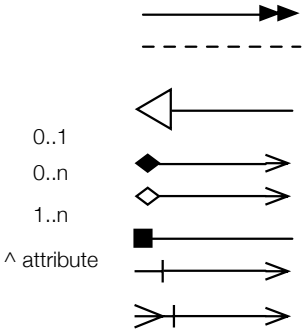
There are 41 productions in simple-bot

```
^io
   ^input-link
         ^rwaves
         ^backward yes/no
         ^forward yes/no
         ^left yes/no
         ^right yes/no
      ^smell
         ^color none/red/blue/purple/...
         ^distance none/0-28
      ^sound silent/left/right/
             forward/backward
      ^clock 1-N
      ^direction north/east/south/west
      ^energy 0-1000
      ^energyrecharger no/yes
      ^health 0-1000
      ^healthrecharger no/yes
      ^missiles 0-N
      ^my-color blue/red/purple/...
      ^radar-distance 1-14
      ^radar-setting 1-14
      ^radar-status on/off
      ^random 0.0-1.0
       ^resurrect no/yes
       ^shield-status on/off
       ^x 1-14
       ^y 1-14
^io
   ^output-link
      ^move.direction left/right/forward/backward/none
      ^rotate.direction left/right
      ^fire.weapon missile
      ^radar.switch on/off
      ^radar-power.setting 1-14
      ^shields.switch on/off
```

Design in Tutorial



SoarML Symbols Template (Omnigraffle)



adds-attribute +

removes-attribute -

replaces-attribute +/-

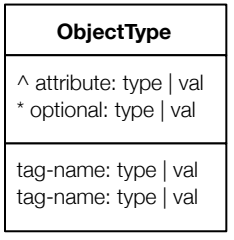
tag attribute

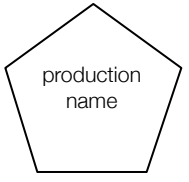
attribute test val

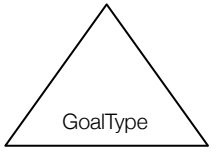
not attribute test val

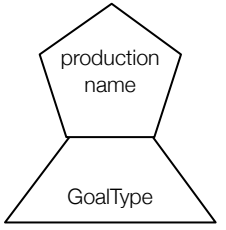
attribute test val AND/OR
attribute test val

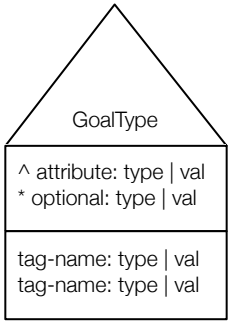
tagged attribute

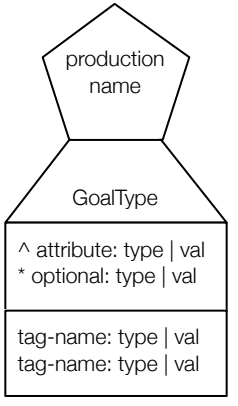


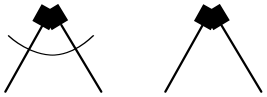


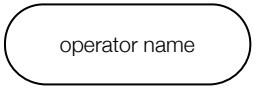


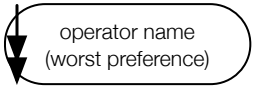


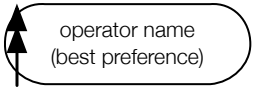


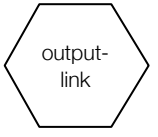


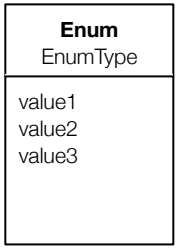


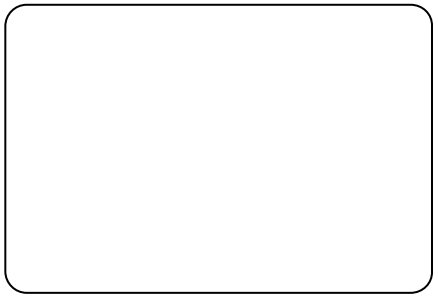




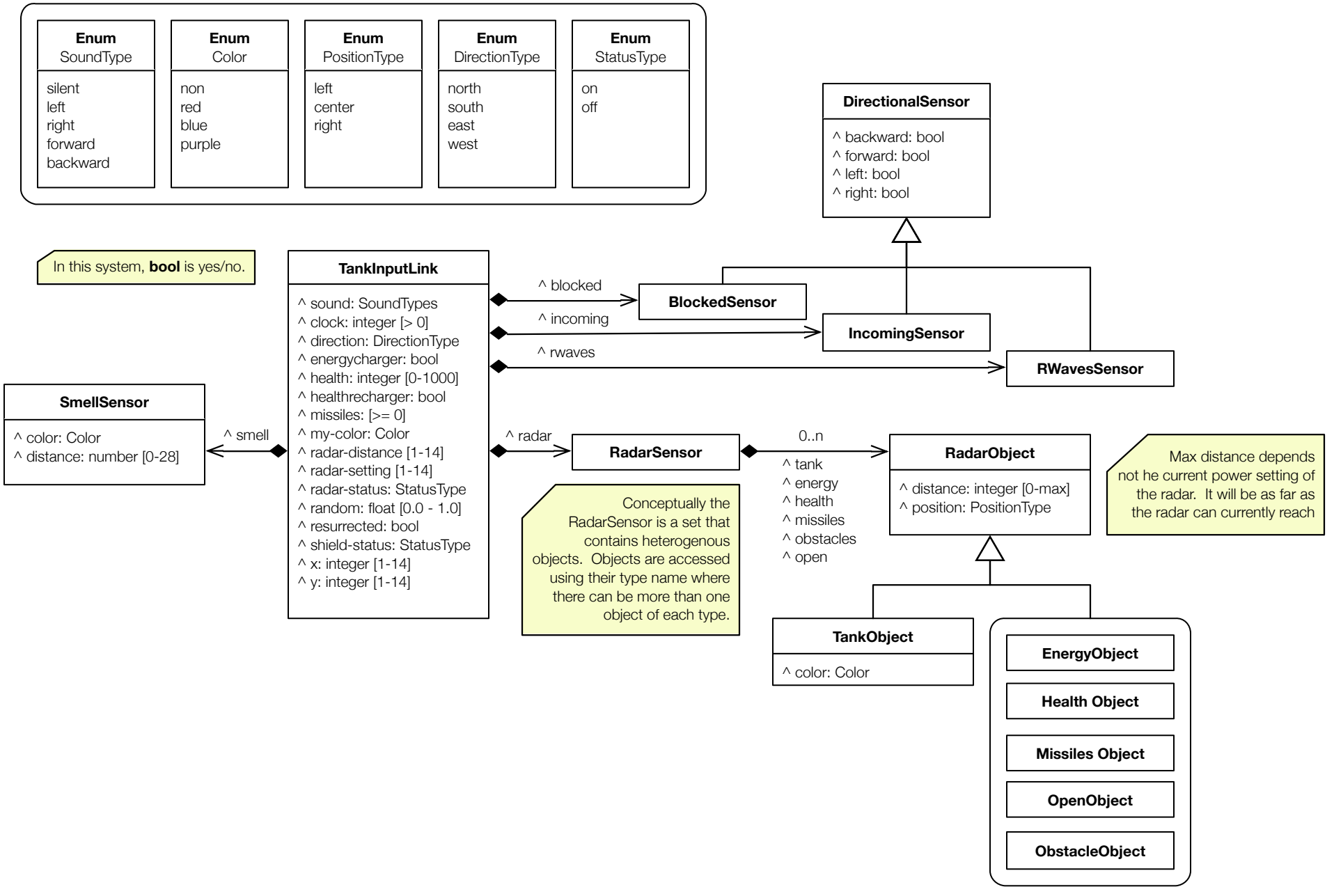




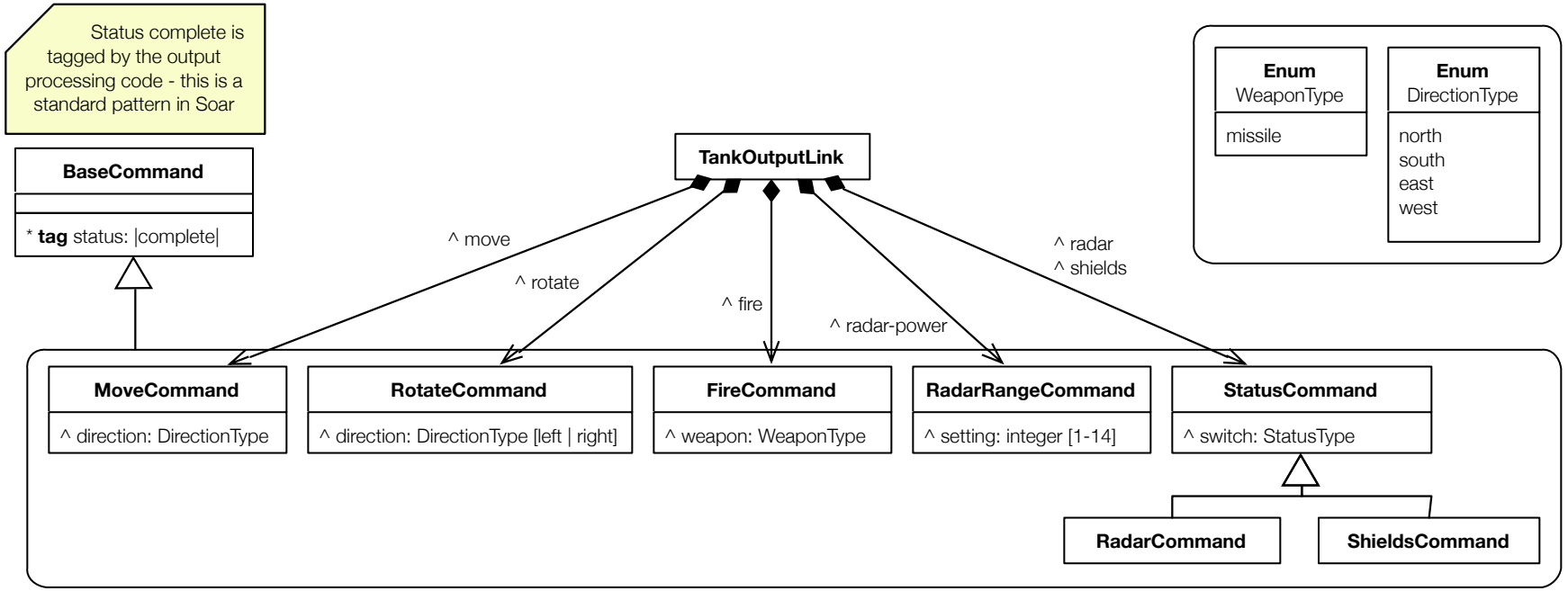




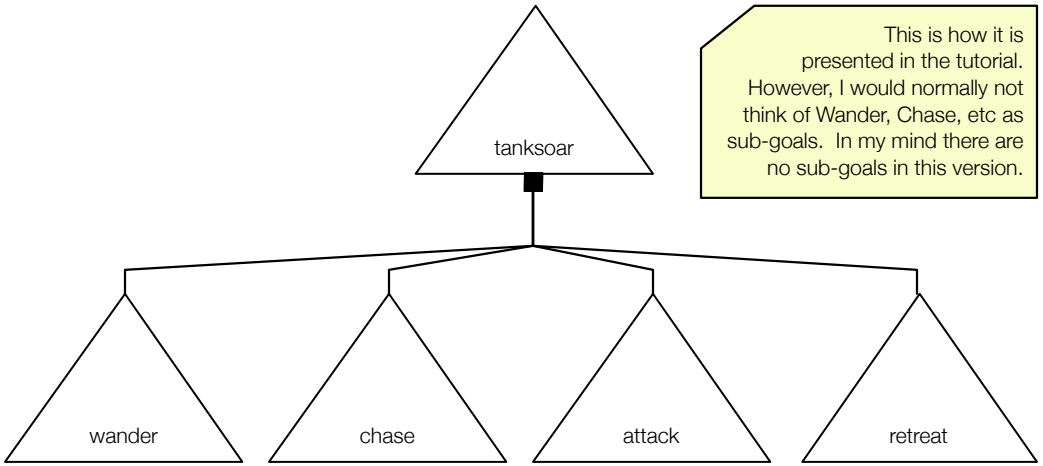
Input Link in SoarML



Output Link in SoarML

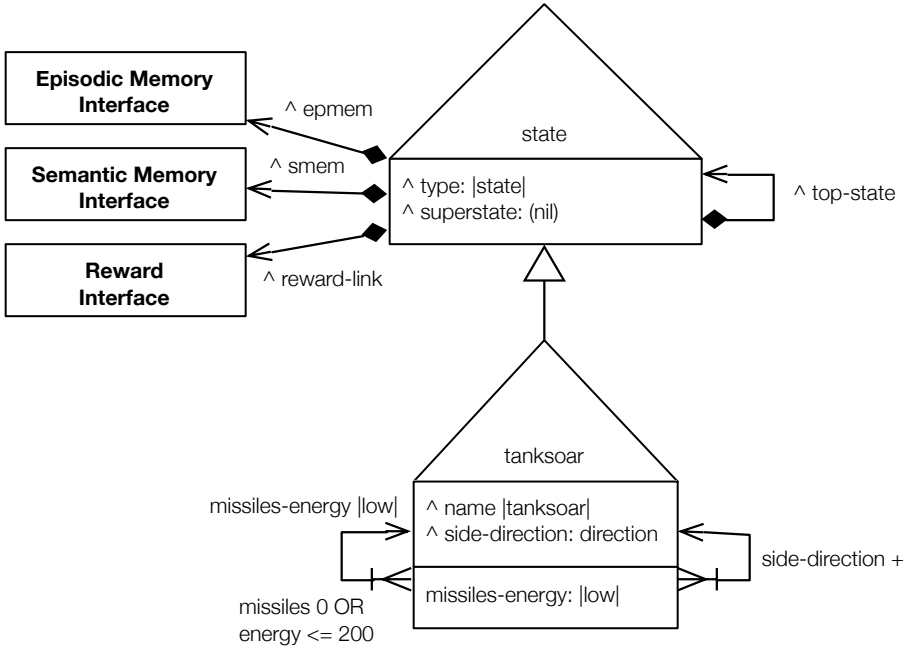


The Goal Hierachy

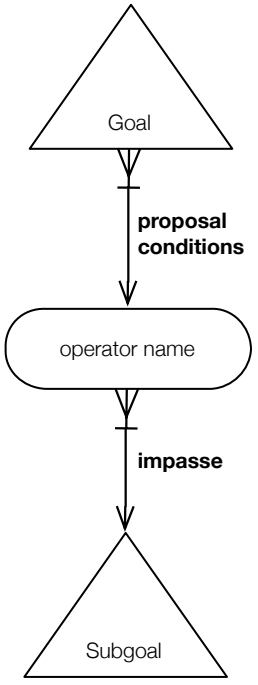


Goals: Conventions

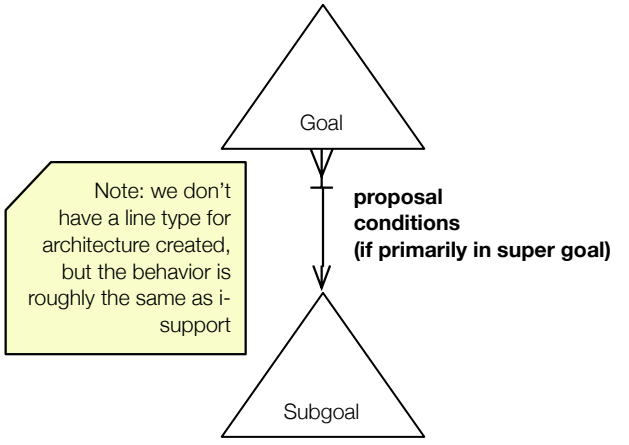
Basic Goal Structure



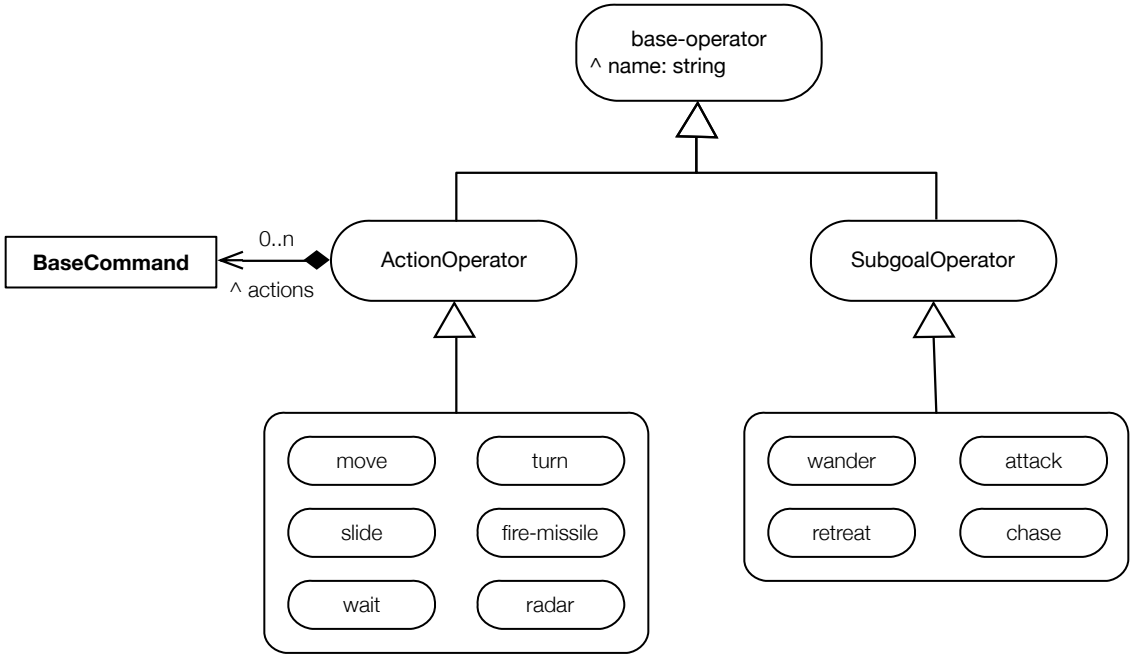
Impasse States as Goals (Michigan Goal System)



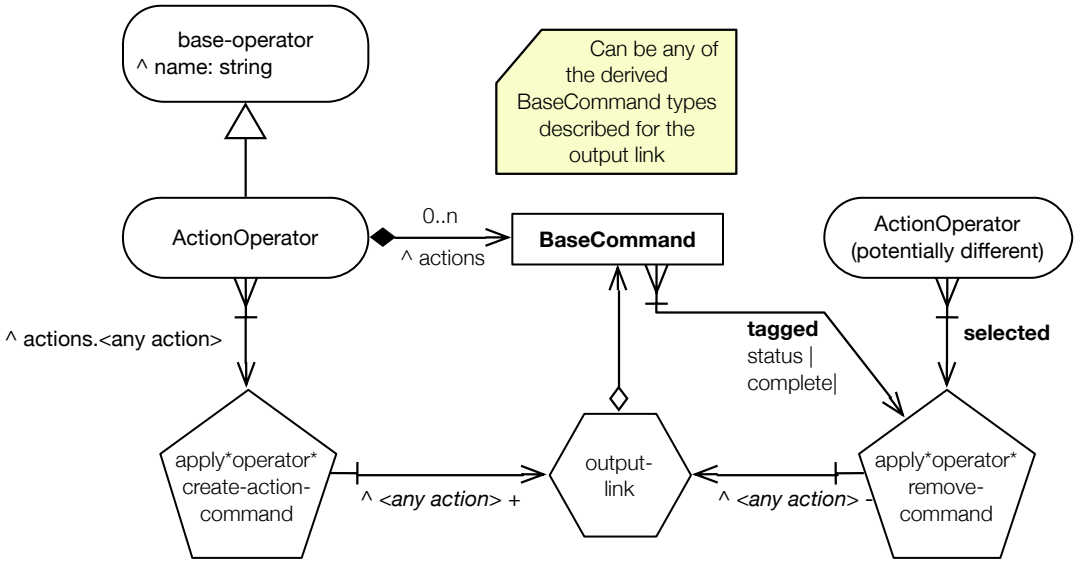
Shorthand



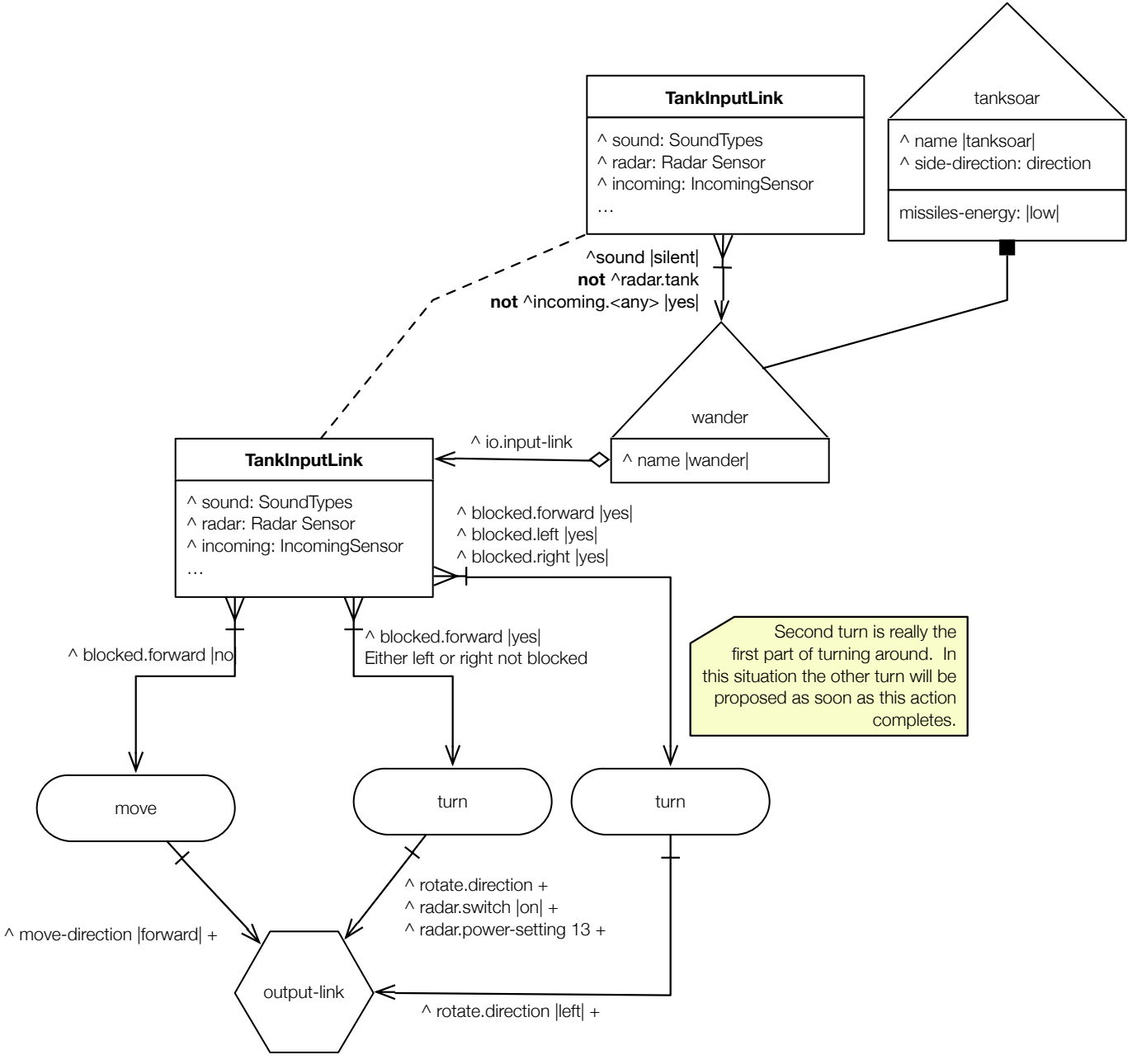
Operator Type Hierarchy



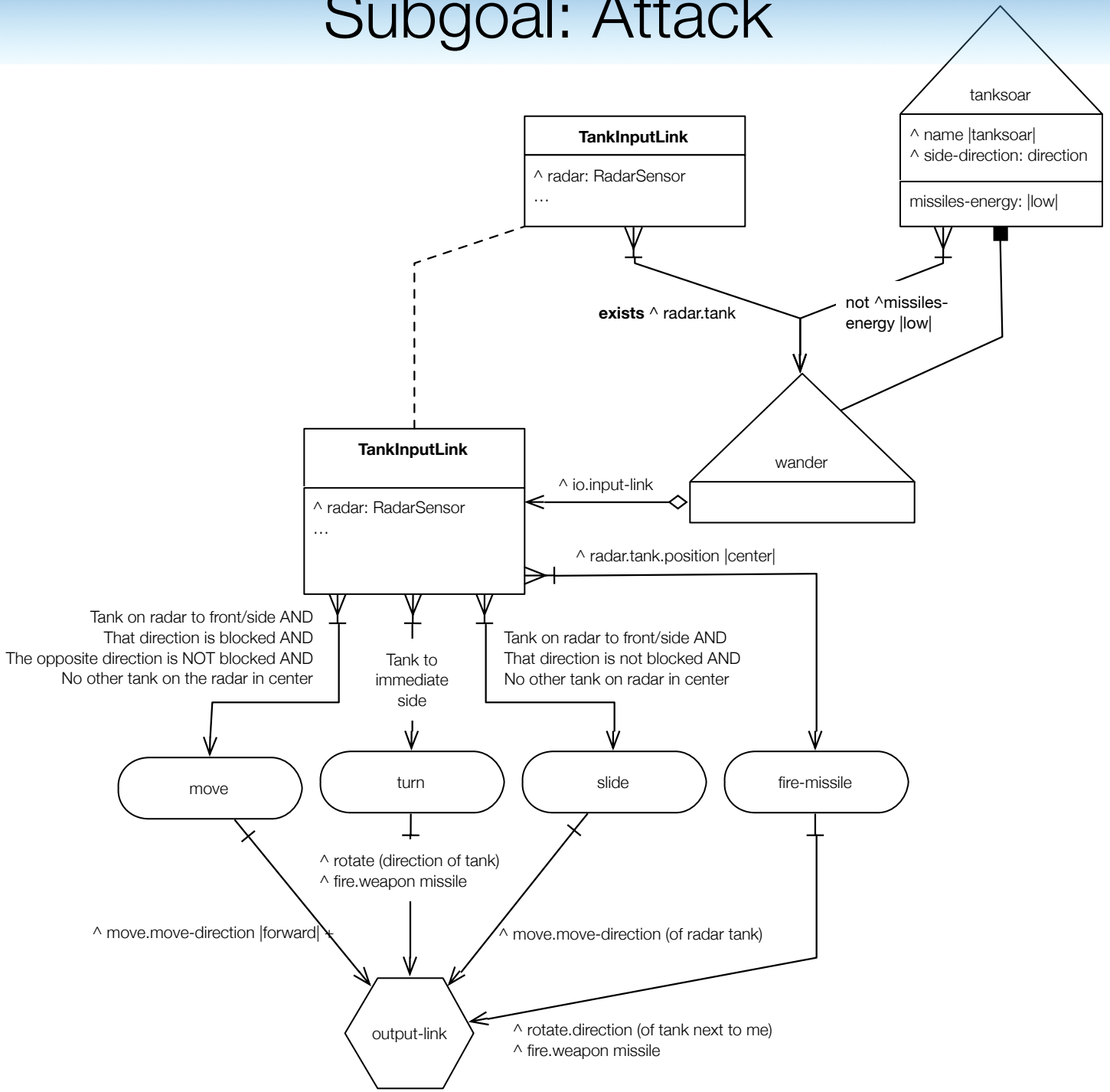
Aspect: External Action Handling



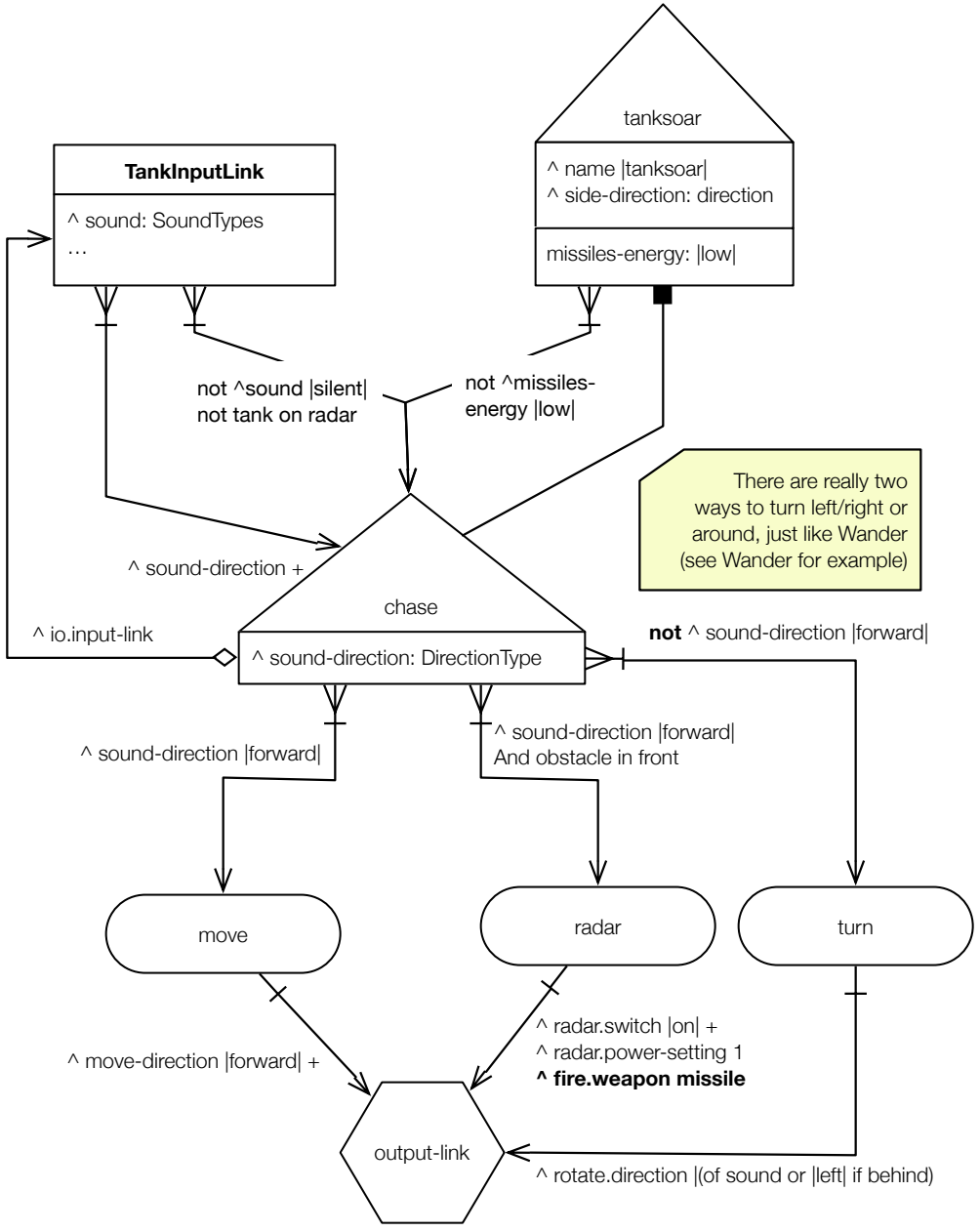
Subgoal: Wander



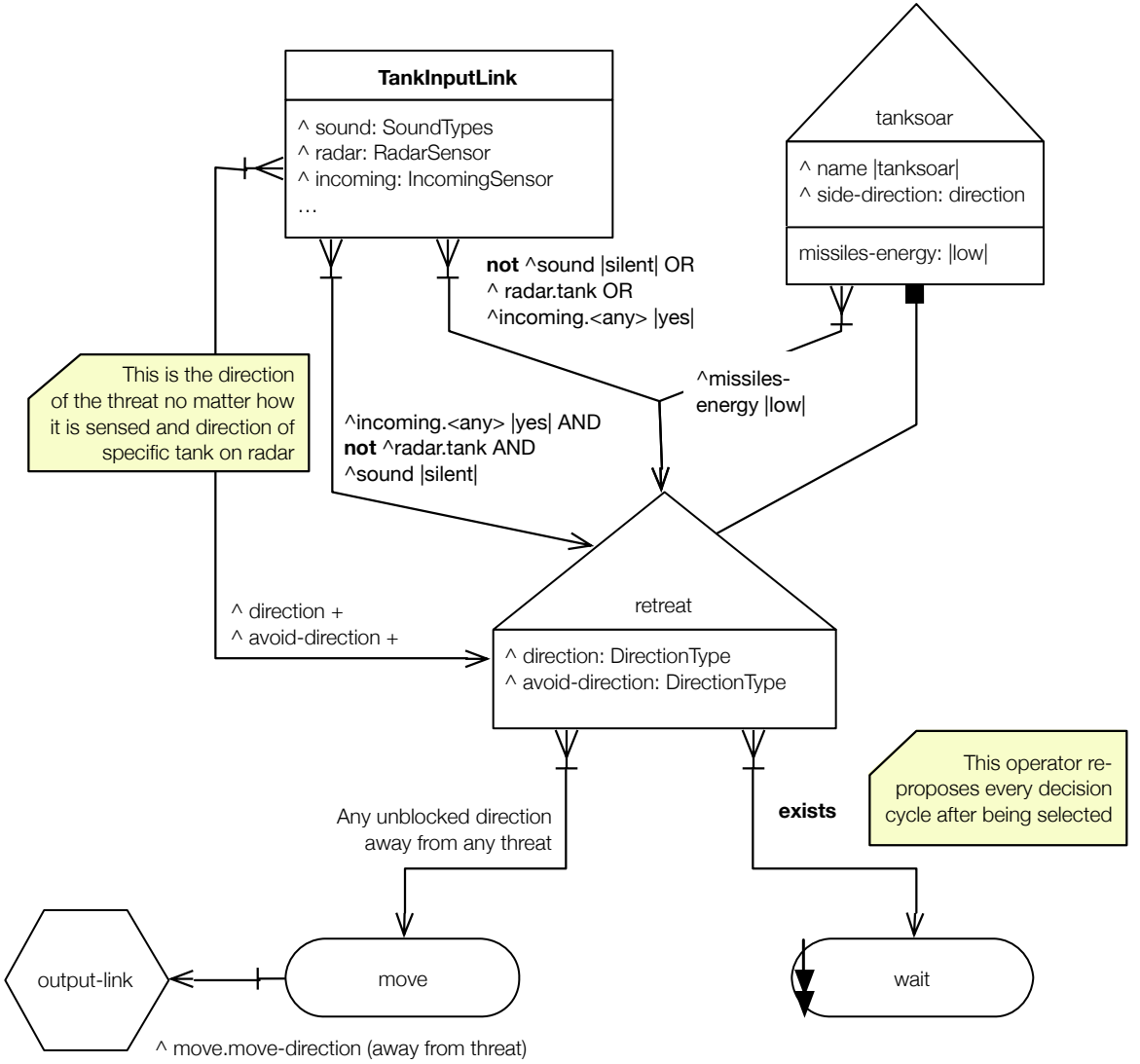
Subgoal: Attack



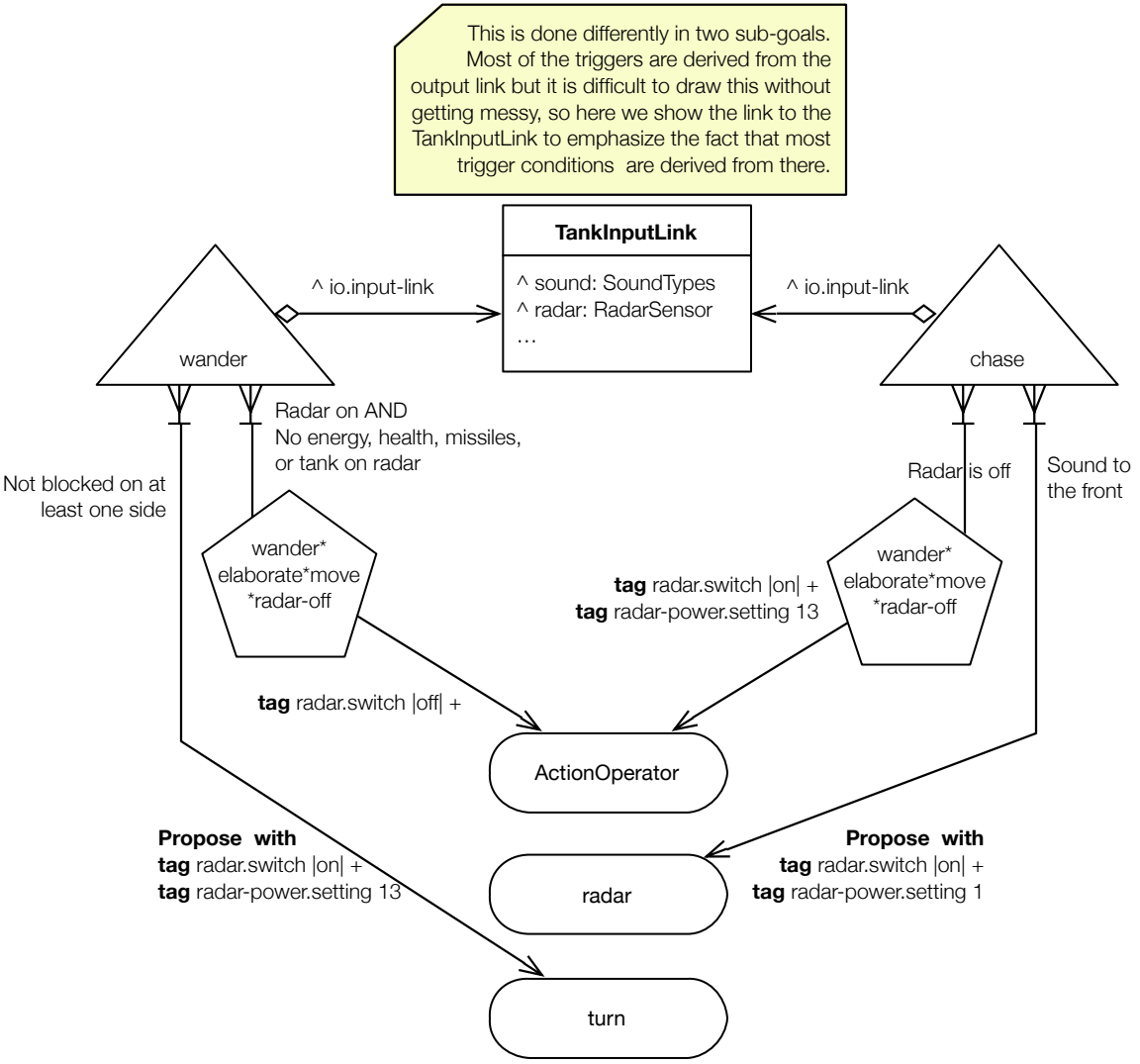
Subgoal: Chase



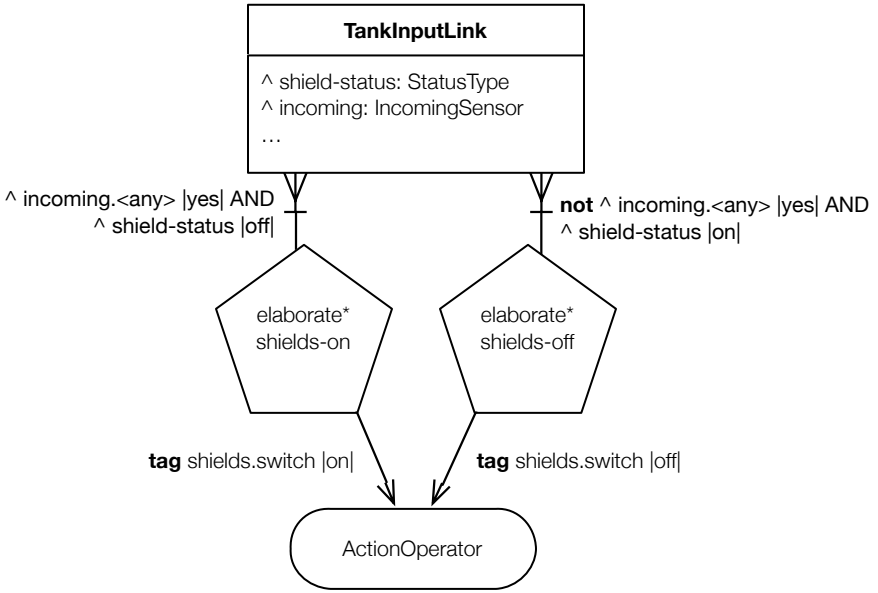
Subgoal: Retreat



Aspect: Radar Control



Aspect: Shield Control



Where to go from here

https://confluence.soartech.com/display/soartechaie/Soar+ML

Useful practice activities

* Documenting existing code
* Whiteboarding a Soar design
* Reviewing design or code

Learning

* Review with Me
* Also: Glenn and Bob M.

Primitive Design Constructs