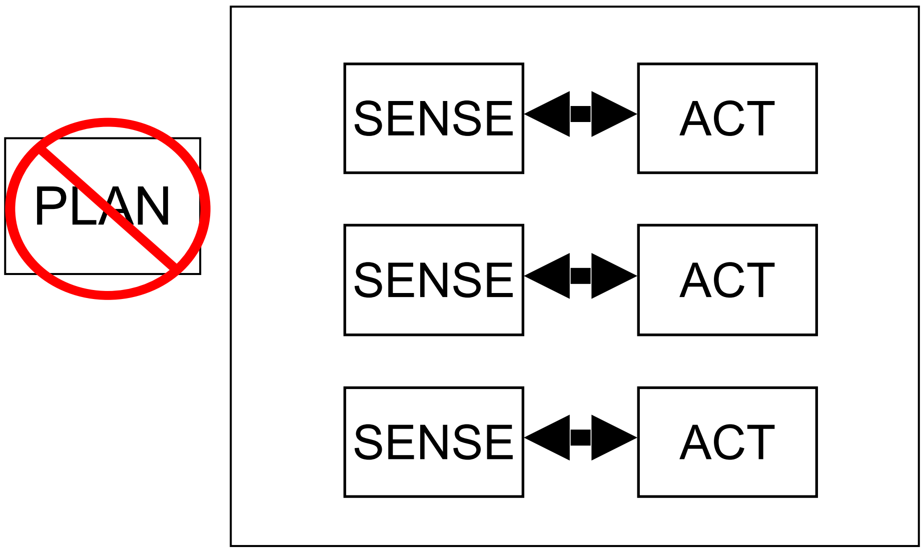
UNIVERSITÀ DEGLI STUDI DI PADOVA

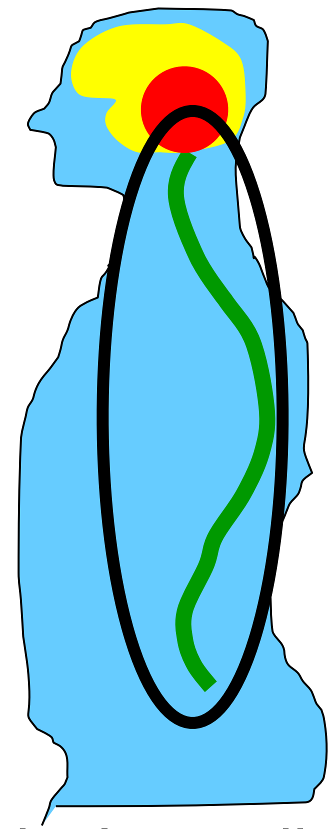
ARCHITECTURES FOR AUTONOMOUS ROBOTS

Prof. Emanuele Menegatti
Intelligent Robotics Course

REACTIVE PARADIGM

Reactive (1986)



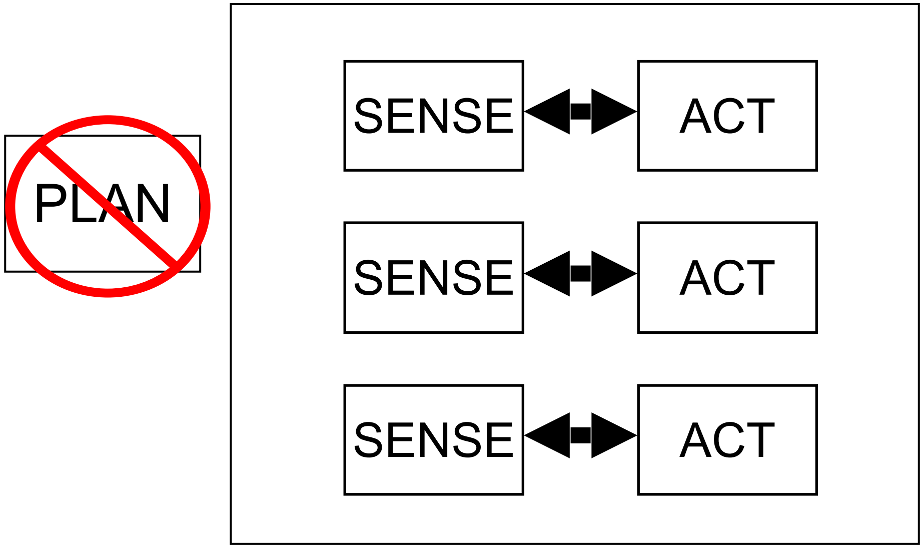


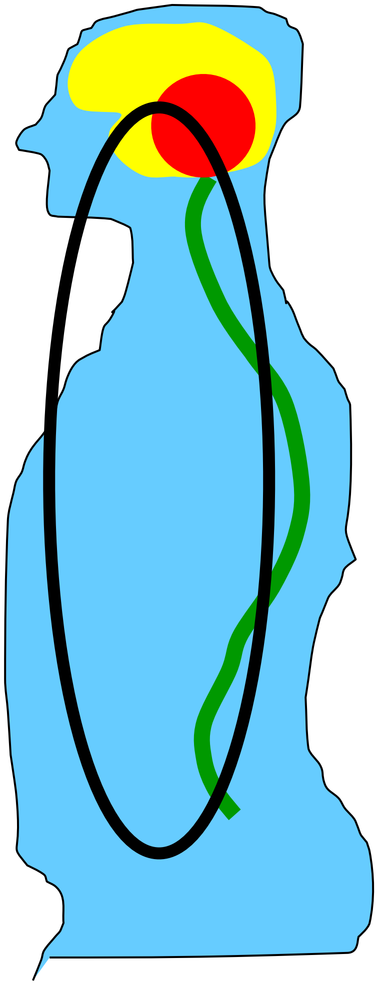
It arises from two reasons:

- Dissatisfaction with the results of the hierarchical paradigm
- The influence of ideas from Winer's ethology and cybernetics

Adapted from R.Murphy

Reactive (1986)





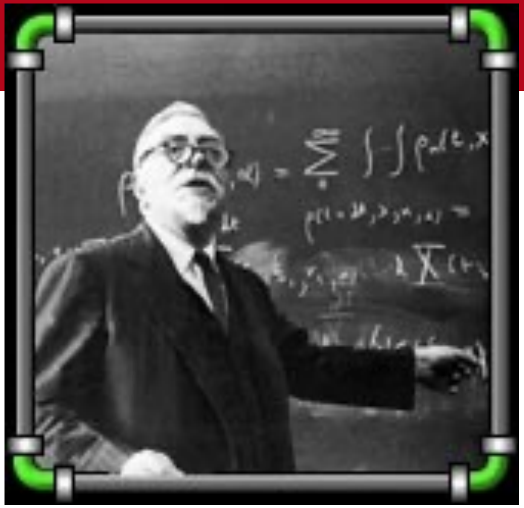
Users loved it because it worked

AI people loved it, but wanted to put PLAN back in

Control people hated it because couldn’t rigorously prove it worked

Adapted from R.Murphy

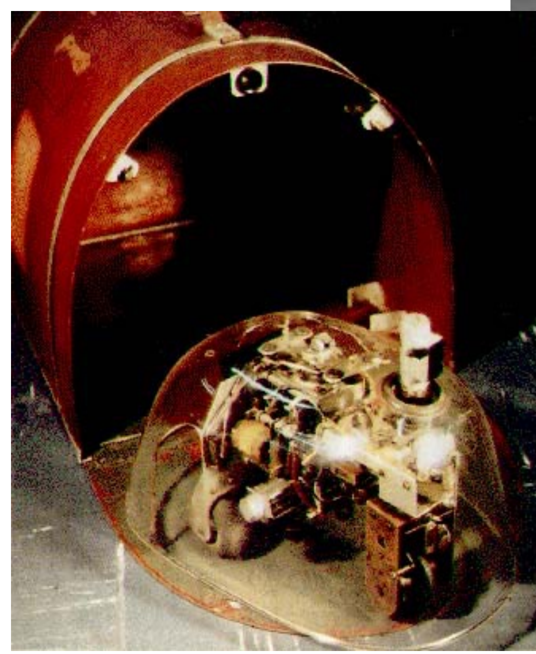
CYBERNETICS



- Cybernetics is a combination of:
  - Control theory;
  - Information theory;
  - Biology
- Try to explain the principles of control in both animals and machines
- Use the math of feedback control systems to express natural behaviors
- The emphasis is on the strong coupling between the organism and its environment
- Initiator of cybernetics: Norbert Wiener in the late 1940s

"Machina Speculatrix" by Grey Walter

- 1950's W. Gray Walter builds the "Machina Speculatrix" (aka "The Turtle") to illustrate how some brain mechanisms work.
- Sensors:
  - Photocells
  - Contact switches
- Behaviors of the turtle:
  - Search for light
  - Attraction towards weak light
  - Repulsion by strong light
  - Rotations and thrusts
  - Battery charging
- Behaviors with priority



Walter's Turtle

- Thrifty: simple is better
- Exploration: constant movement to avoid traps
- Attraction: movement towards positive stimuli
- Aversion: turning away from negative stimuli
- Recharging: it could return to a predefined spot for recharging

Complex behaviours arise from the combination of these simple actions

Slide di M. Reggiani

What is a Behavior

A Mapping of sensory inputs to a pattern of motor actions that are used to achieve a task. We can classify three broad categories of behaviors:

- –Reflexive behaviors:
  - Stimulus-response
  - Hard-wired for fast response
    - Example: (physical) knee-jerk reaction
- –Reactive behaviors:
  - Learned
  - “Compiled down” to be executed without conscious thought
    - Examples: “muscle memory” –playing piano, riding bicycle, running, etc.
- –Conscious behaviors:
  - Require deliberative thought
    - Examples: writing computer code, completing your tax returns, etc.

BRAITENBERG VEHICLES

- Valentino Braitenberg, proposed studying principles of intelligence by building a series of agents of increasing complexity, the Braitenberg vehicles.

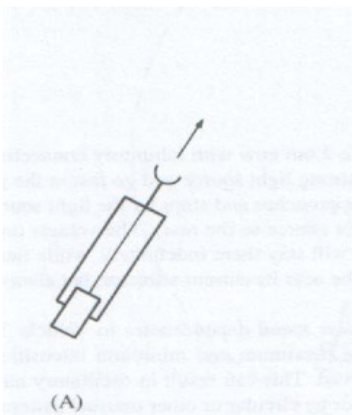
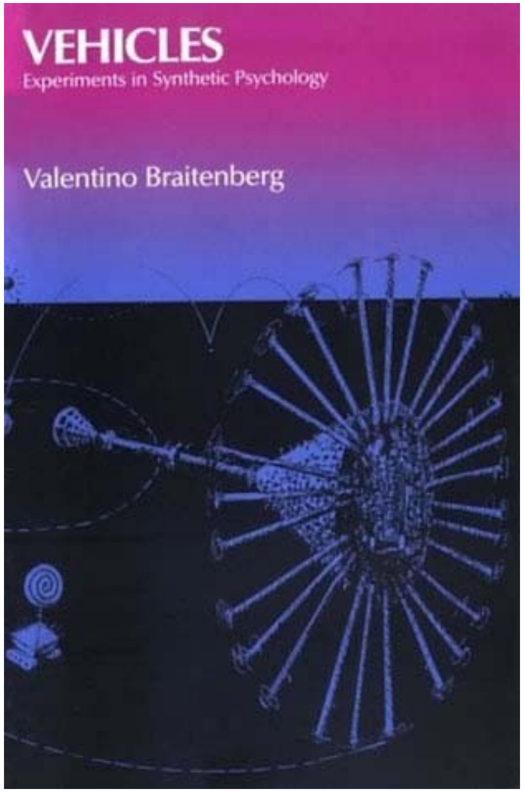


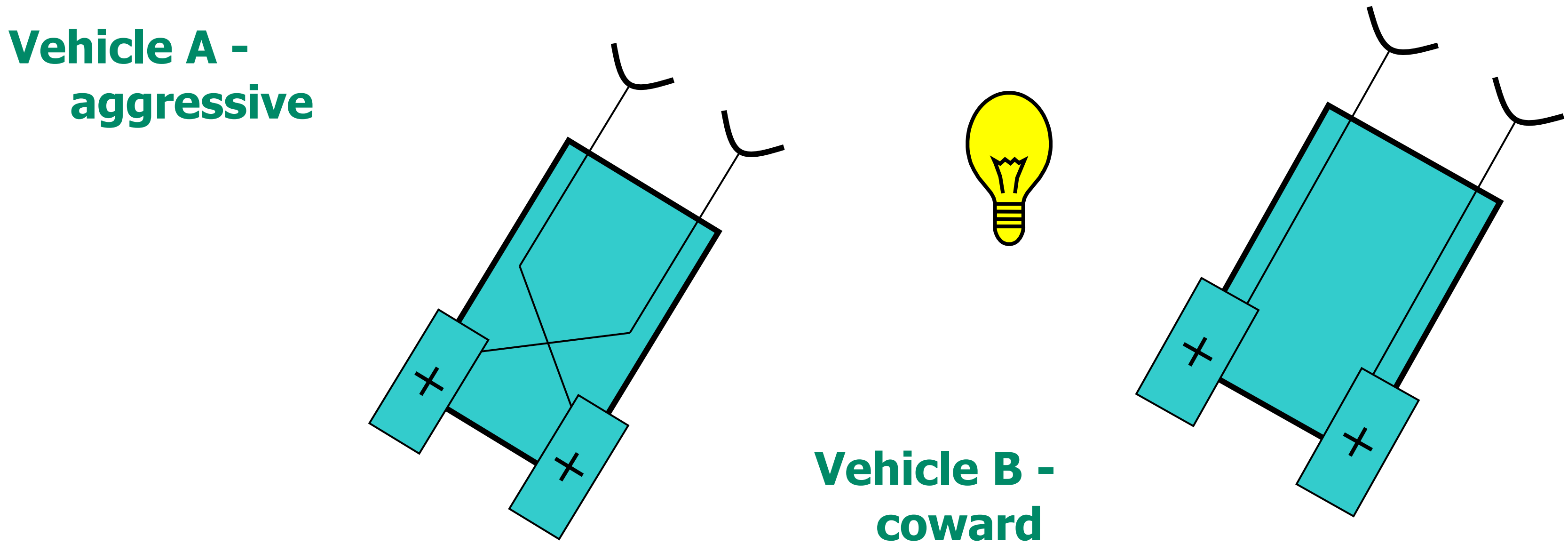
Figure 1.6
Braitenberg Vehicles

Each vehicle has sensors and motors. Vehicle No. 1 has one sensor and one motor. The more the sensor augment its input, the faster the motor goes.



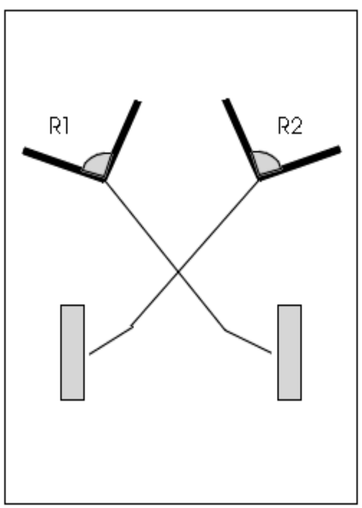
V. Braitenberg Vehicles: Experiments in synthetic psychology The MIT Press, 1984

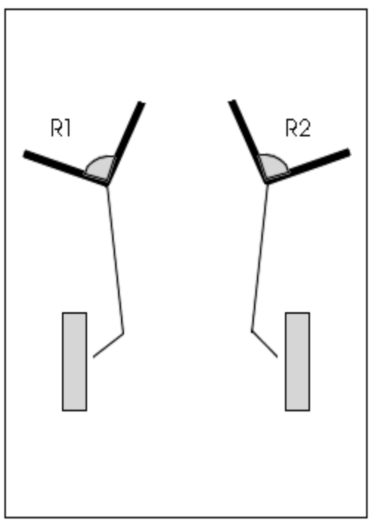
Vehicle Type No. 2: coward or aggressive

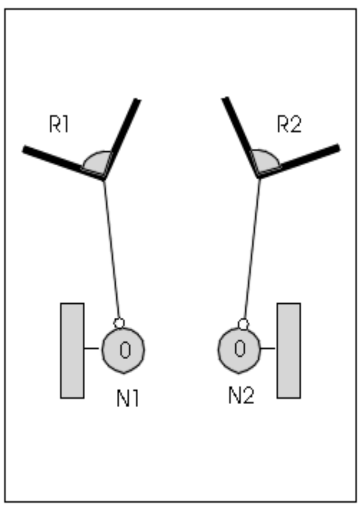


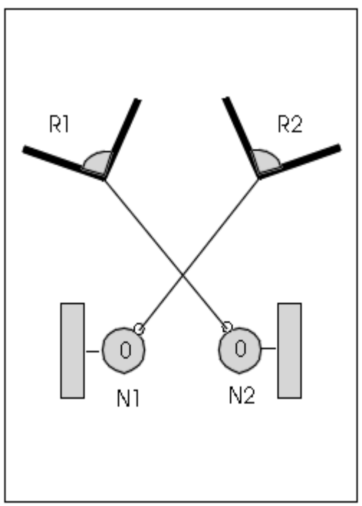
- Braitenberg Vehicle No. 2 has two sensors, one for each side, and two motors
- There are two ways to connect sensors with motors
- Because the right sensor of Vehicle A is closer to the light, the left motor gets more stimulation, and then Vehicle A is turning towards the light (aggressive)
- Because the left sensor of Vehicle B is closer to the light, the left motor gets more stimulation, and then Vehicle B is escaping from the light (coward)

- We can add motors, and connect them to sensors in different ways.
- Sensors can either stimulate or inhibit motors activation.
- We can make motor dependency on sensors according to a non linear law.
- We can allow thresholds in powering a motor by sensor stimulus.







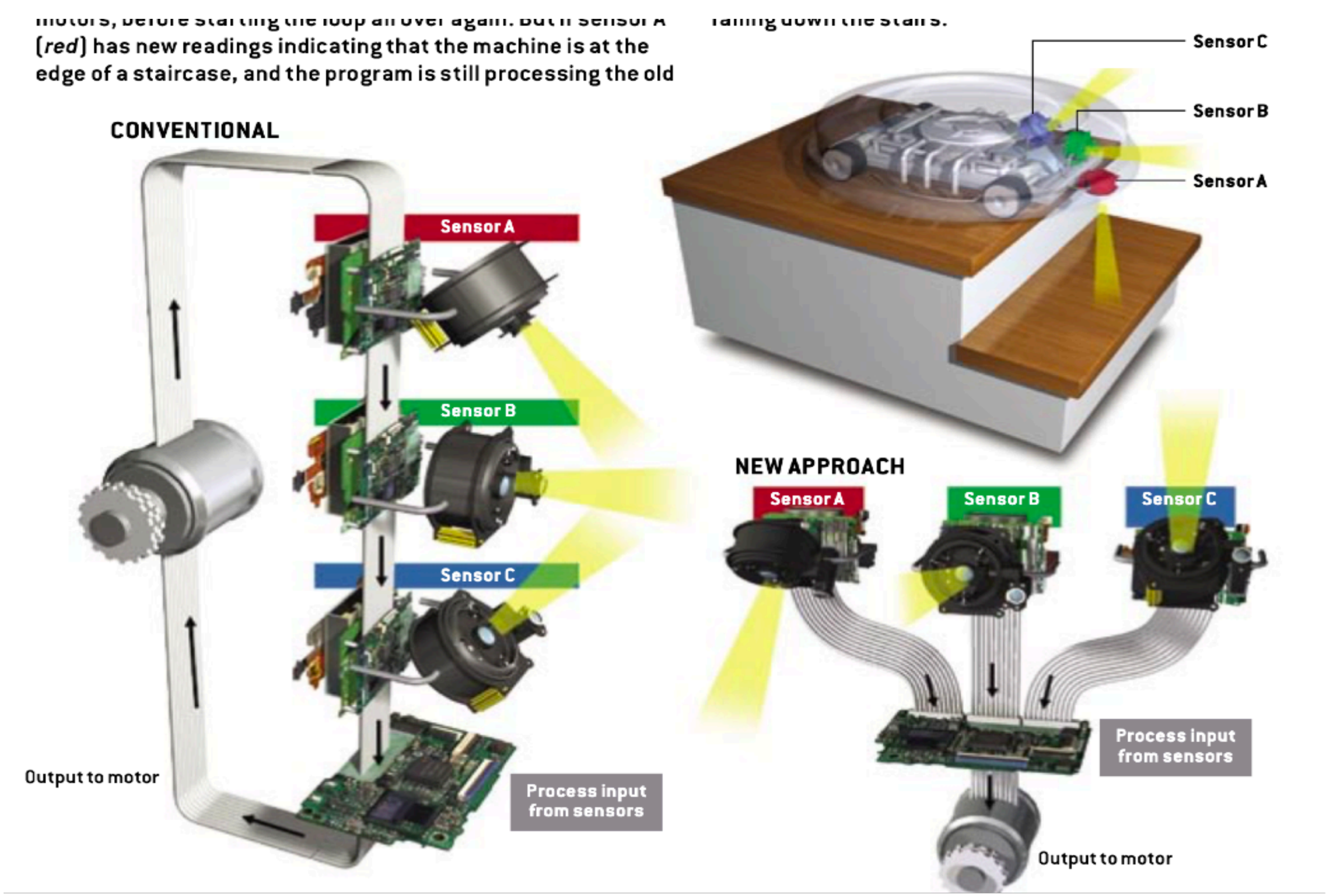


Segmenting the behaviors

- Braitenberg introduced 14 different types of vehicles. The more complex ones have been augmented by adding more sophisticated types of neural networks.
- The simplest ones, from No. 1 to No. 6, base largely their behavior control on a direct sensor-motor coupling. The most complex one, from No. 7 till No. 14, have a strong cognitivistic flavor
- The simplest types are the most interesting, because they illustrate the sensory-motor coordination,

- There are no Internal Actions. Processes are only feeding some state variables.
  – Segmentation of behaviors is hard to solve! Vehicles have no internal actions corresponding to some specific observer-based behavioral categories
- What the agent really does is engaging in behaviors that are emergent from the dynamics of its internal variables (sensor stimulation and motor speed) determined by the interaction of the vehicle with its environment.

THE FLOW OF PERCEPTION

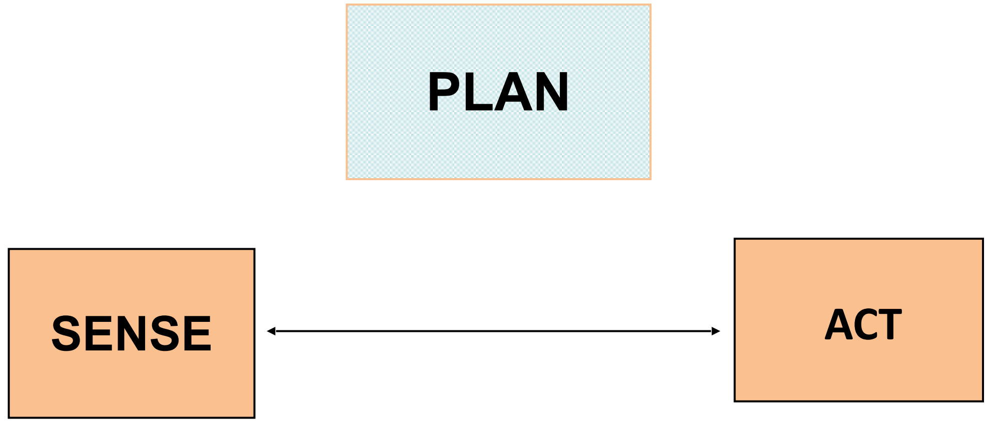


GOOFAI architecture **MODERN architecture**

- Behavior: relationship between sensorial inputs and motor behavior to accomplish a task or (equivalent) to react to a stimulus

- A Reactive Robot closely links perception to action without any abstract representation or temporal history

- Reactive Paradigm



REACTIVE PARADIGM

Primitive	Input	Output
SENSE	Sensor data	Sensed Information
ACT	Sensed information or directives	Actuator commands

- Uses multiple instances of couples SENSE-ACT called behaviors that get turned on/off based on stimulus

- No PLAN

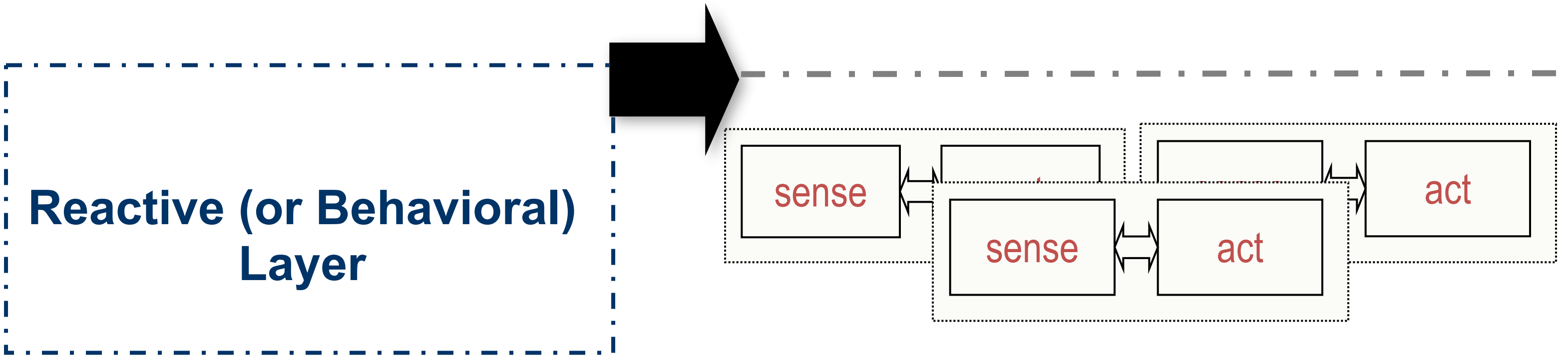


DIAGRAM STIMULUS-RESPONSE

The most intuitive and least formal of the methods of expression

What is a Behavior

A Mapping of sensory inputs to a pattern of motor actions that are used to achieve a task. We can classify three broad categories of behaviors:

- –Reflexive behaviors:
  - Stimulus-response
  - Hard-wired for fast response
    - Example: (physical) knee-jerk reaction
- –Reactive behaviors:
  - Learned
  - “Compiled down” to be executed without conscious thought
    - Examples: “muscle memory” –playing piano, riding bicycle, running, etc.
- –Conscious behaviors:
  - Require deliberative thought
    - Examples: writing computer code, completing your tax returns, etc.

REACTIVE SYSTEMS: Characteristics

- Behaviours serve as elementary blocks to build robotic actions
- In generating an answer, the use of abstract representation of knowledge is avoided
- They are inherently modular systems from a software point of view

Key Questions

- What are the *appropriate behavior* to build robotic systems?
- What a *primitive behavior* really is?
- How can we coordinate efficiently these behaviors?
- How are behaviors connected to sensors and actuators?

NO UNIVERSALLY ACCEPTED ANSWERS…

EXAMPLES OF BEHAVIORS/1

- ***Exploration/directional behaviors*** *(move in a general direction)*
  - heading based
  - wandering
- ***Goal-oriented appetitive behaviors*** *(move towards an attractor)*
  - discrete object attractor
  - area attractor
- ***Aversive/protective behaviors*** *(prevent collisions)*
  - avoid stationary objects
  - elude moving objects (dodge, escape)
  - aggression
- ***Path following behaviors*** *(move on a designated path)*
  - road following
  - hallway navigation
  - stripe following

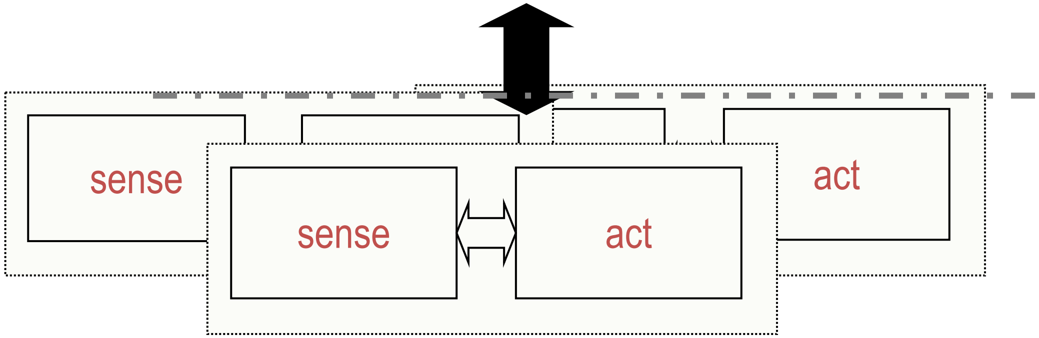
EXAMPLES OF BEHAVIORS/2

- ***Postural behaviors***
  - balance
  - stability
- ***Social/cooperative behaviors***
  - sharing
  - foraging
  - flocking/herding
- ***Teleautonomous behaviors*** (coordinate with human operator)
  - influence
  - behavioral modification
- ***Perceptual behaviors***
  - visual search
  - ocular reflexes

EXAMPLES OF BEHAVIORS/3

- ***Walking behaviors*** *(for legged robots)*
  - gait control
- ***Manipulator-specific behaviors*** *(for arm control)*
  - reaching
- ***Gripper/dextrous hand behaviors*** *(for object acquisition)*
  - grasping
  - enveloping

Control Theory is "Lower Level" But Doesn't Necessary Capture it All



Reactive (inner loop control, behaviors):

- Tightly coupled with sensing, so very fast
- Many concurrent stimulus-response behaviors, strung together with simple scripting with FSA
- Action is generated by sensed or internal stimulus
- No awareness, no mission monitoring
- Models are of the vehicle, not the "larger" world

Adapted from R.Murphy

- Focus in AI from 1986-1996

- Things you can do with behavioral robotics
  - Roomba, Aibo
  - Guarded motions (don’t hit anything), panic behaviors (stop!), “Macros” such as self-righting in teleoperation

- Advantages
  - Direction perception is usually simple
  - High modularity, add new behaviors without reprogramming old working behaviors, degrades gracefully (depending on the technical architecture)

Subsumption Architecture

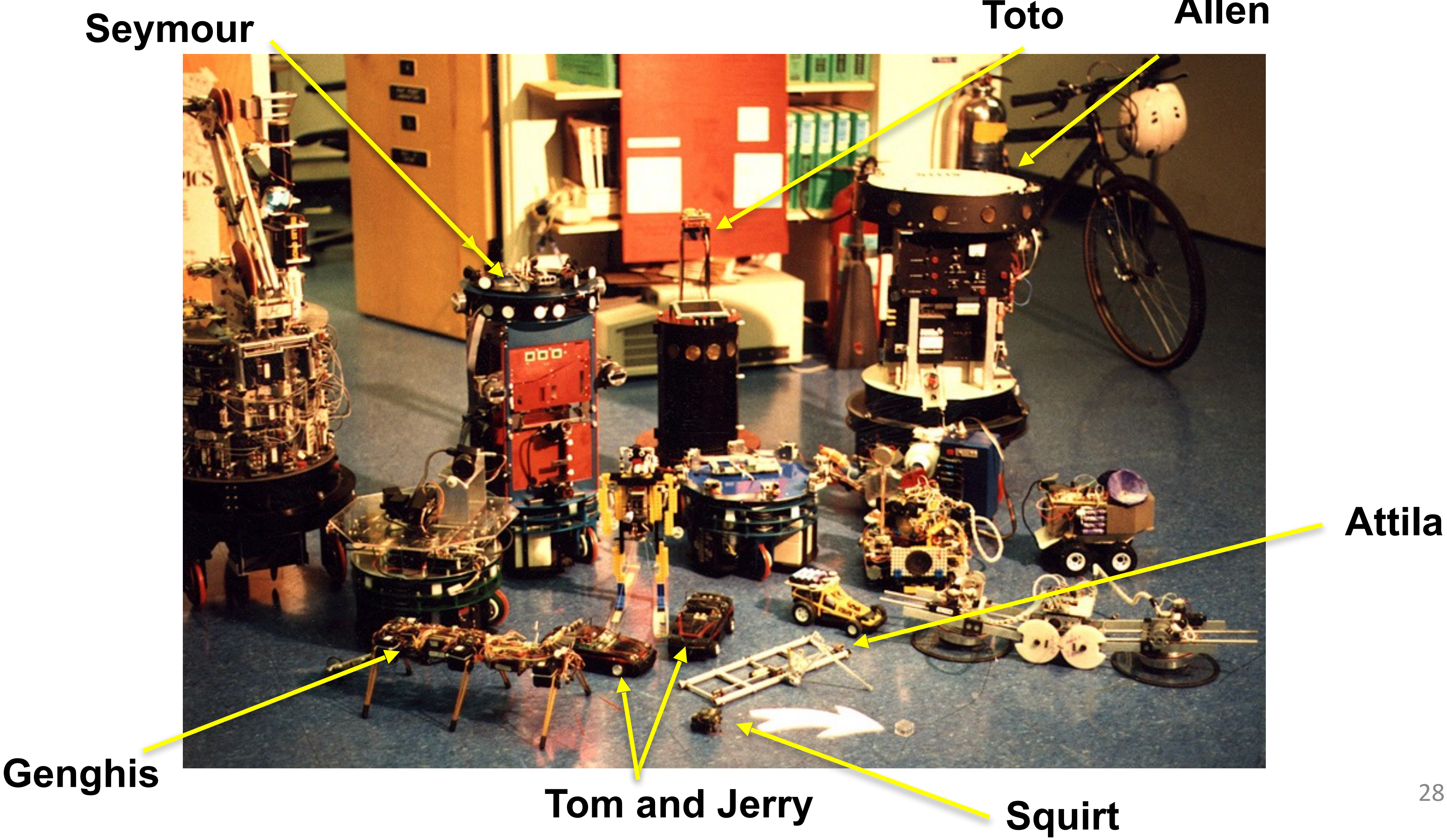
- Rod Brooks (MIT-1986)

SUBSUMPTION ARCHITECTURE

Name	Subsumption architecture
Background	First reactive architetture
Precursor	Braitenberg 1984; Walter 1953; Ashby 1952
Main methodology design	Experimental
Developer	Rodney Brooks (MIT)
Response coding	Mostly discrete
Method of coordination	Competitive (priority-based arbitration with inhibition or suppression)
Method of programming	AFSMs and later Behavior Language
Robot	Allen, Genghis, Squirt, Toto, Seymour, Polly
Reference	Brooks 1986: A robust layered control system for a mobile robot Brooks 1990: Elephants Don't Play Chess

SUBSUMPTION ARCHITECTURE

Some of the robots used in the Laboratory of Rod Brooks at MIT to test his subsumpsion architecture



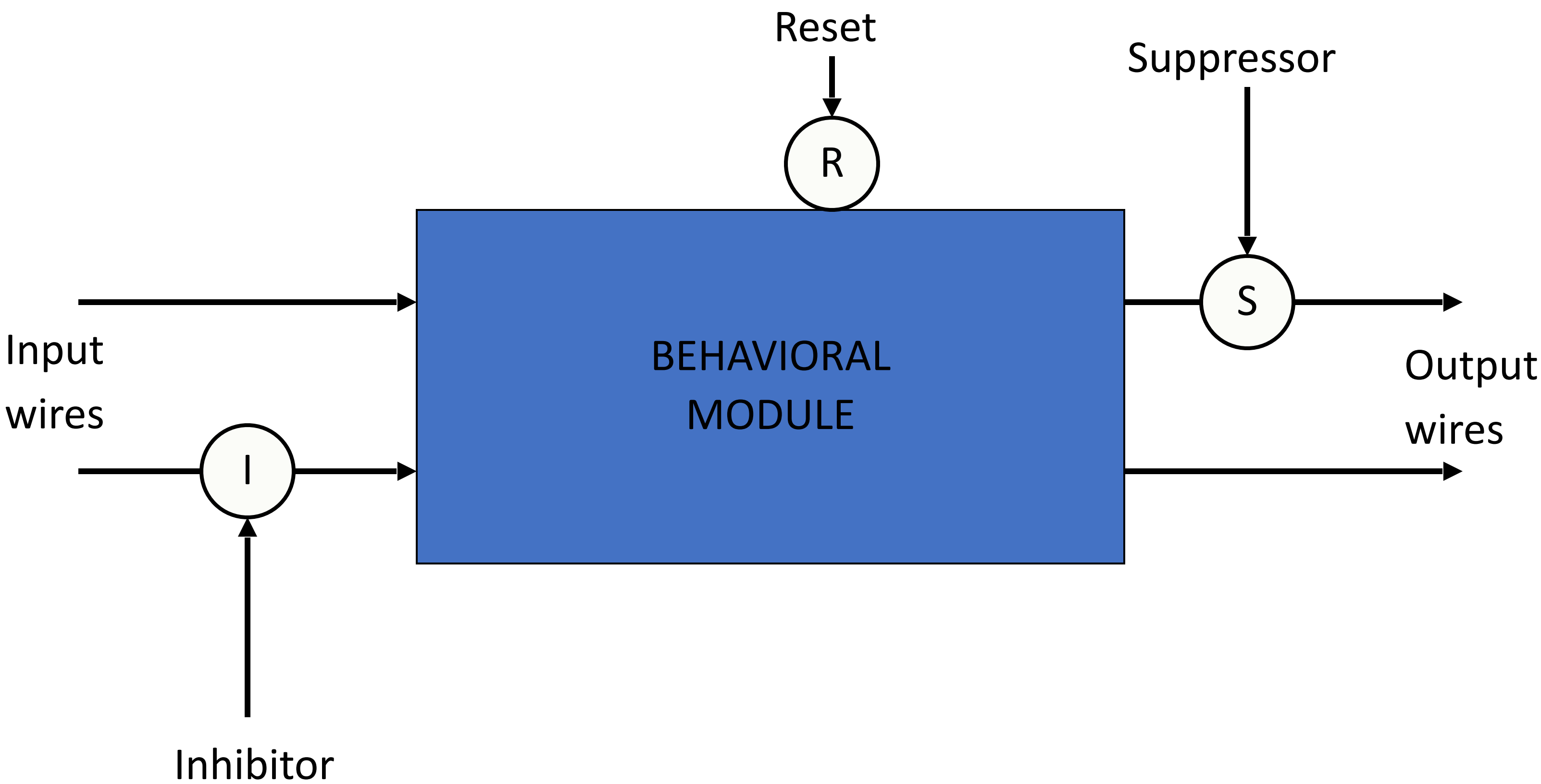
SUBSUMPTION ARCHITECTURE: Principles

- Complex behaviors are not necessarily the product of a complex control system
- *"Intelligence is in the eye of the observer"*
  (*Intelligence without Reasoning*)
- *"The world is the best model of itself"*
  (*Intelligence without Representation*)
- Simplicity is a virtue – Robots should be cheap
- Robustness in the presence of a noise or a sensor breakage is one of the design goals
- *"Planning is just a way of avoiding deciding what to do next"*
- Onboard processing is important
- Systems should be built incrementally
- No representation. No calibration. No complex computers. No broadband communication

- Decision making by a set of task accomplishing behaviors
  - Behaviors are direct mappings from states to actions
    - Processing of raw sensor data
    - Direct coupling between state and action,
      e.g. light switch pressed → light on
    - Behaviors implemented as asynchronous finite state machines

- Mechanism for action selection: *subsumption hierarchy*
  - Behaviors are organized in layers
    - Behaviors “fire” simultaneously
    - Higher layer behaviors inhibit lower level ones, i.e. higher level behaviours SUBSUME lower level behaviours

Each behavior is represented by an Augmented Finite State Machine” (AFSM)



ASSEMBLING BEHAVIORS

- Because there is no central control, AFSMs communicate with each other via inhibition and suppression signals.
- Inhibition signals block signals from reaching actuators or AFSMs, and suppression signals blocks or replaces the inputs to layers or their AFSMs.
- This system of AFSM communication is how higher layers **subsume** lower ones (see next figures), as well as how the architecture deals with priority and action selection arbitration in general

- The development of layers follows an intuitive progression.
- First, the lowest layer is created, tested, and debugged.
- Once that lowest level is running, one creates and attaches the second layer with the proper suppression and inhibition connections to the first layer.
- After testing and debugging the combined behavior, this process can be repeated for (theoretically) any number of behavioral modules

ASSEMBLING BEHAVIORS

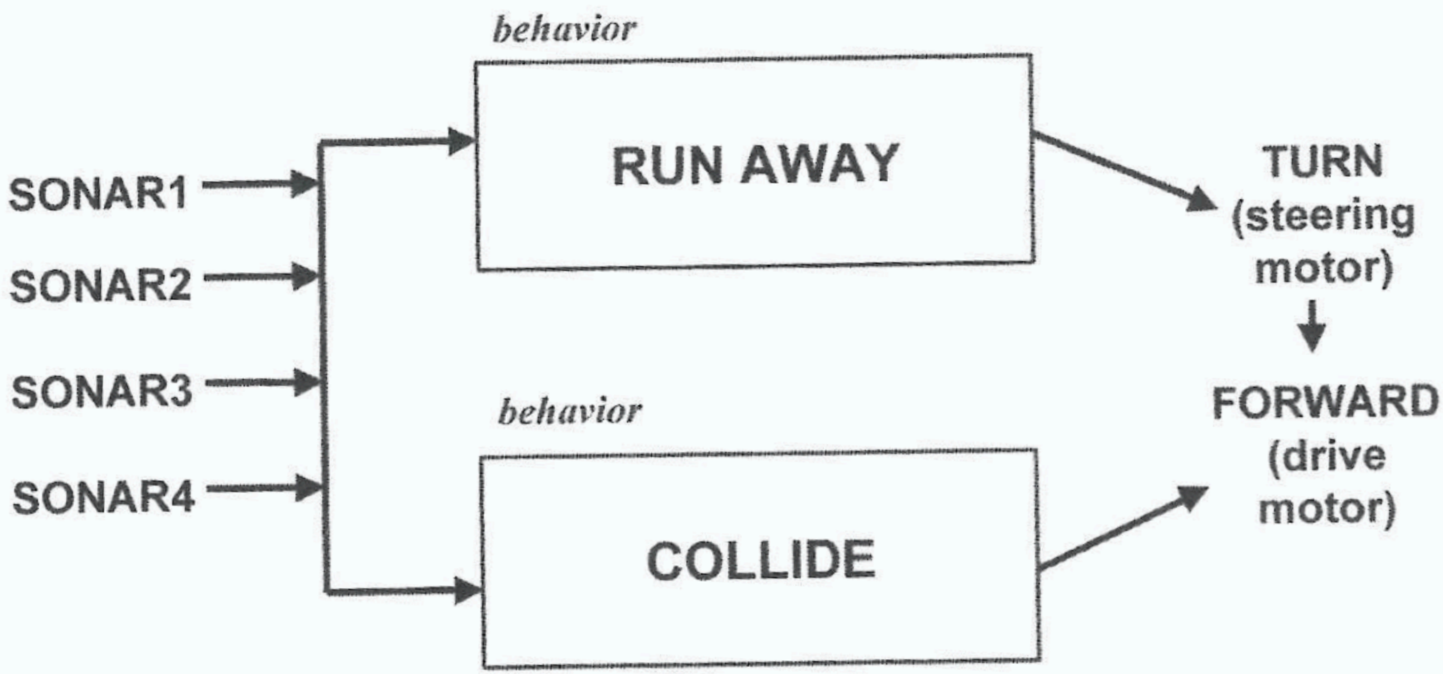


Figure 8.15 Level 0 recast as primitive behaviors.

ASSEMBLING BEHAVIORS

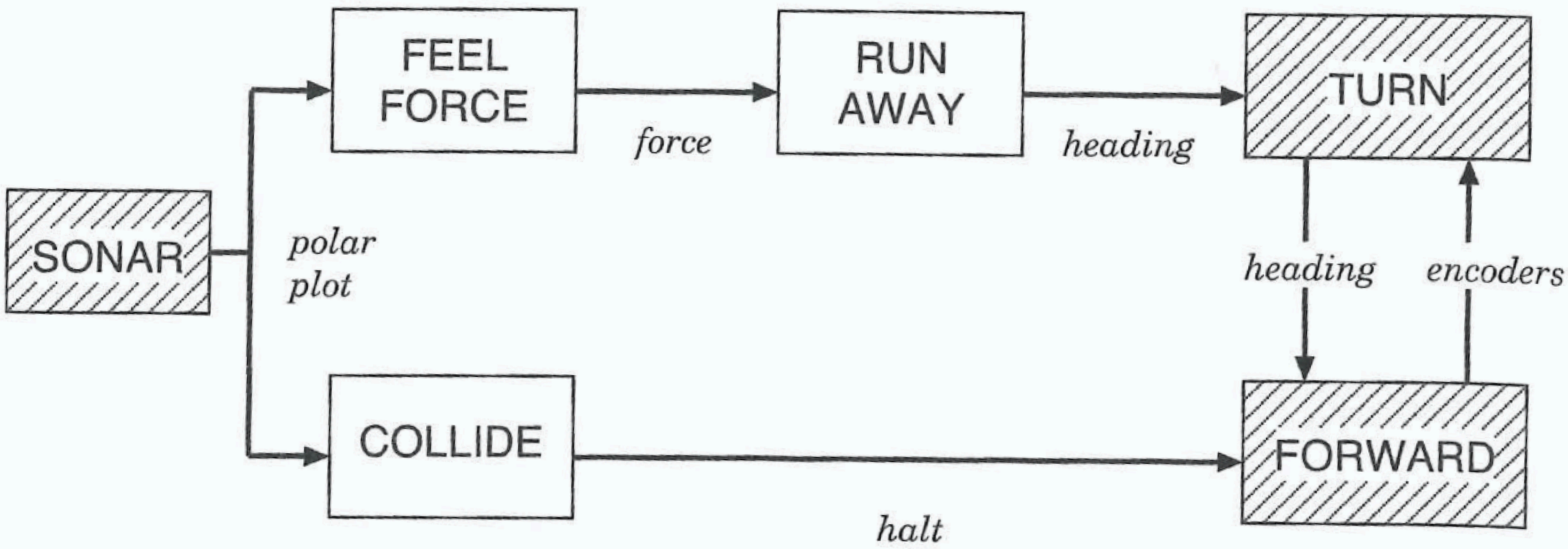


Figure 8.13 Level 0 in the subsumption architecture.

ASSEMBLING BEHAVIORS

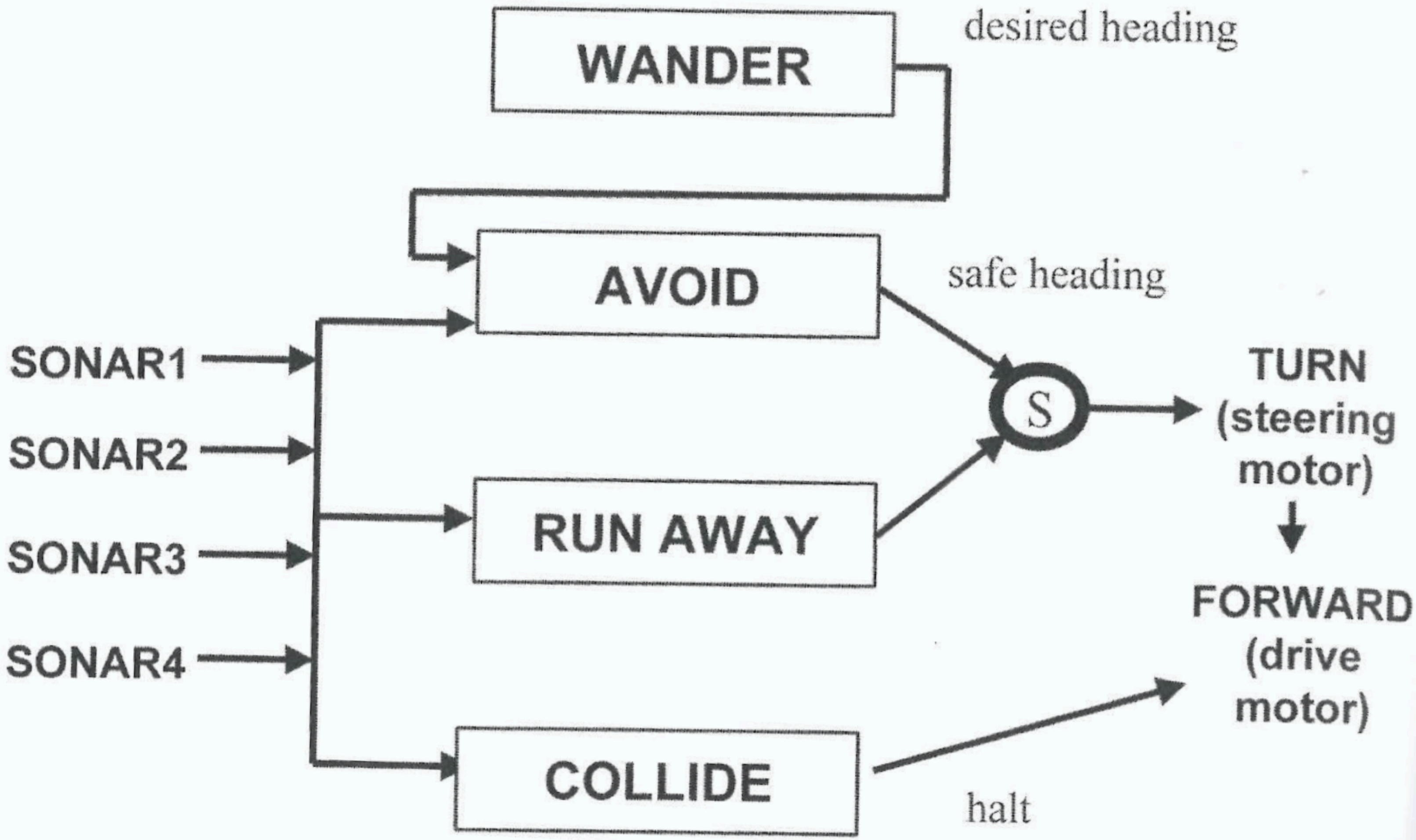


Figure 8.17 Level 1 recast as primitive behaviors.

ASSEMBLING BEHAVIORS

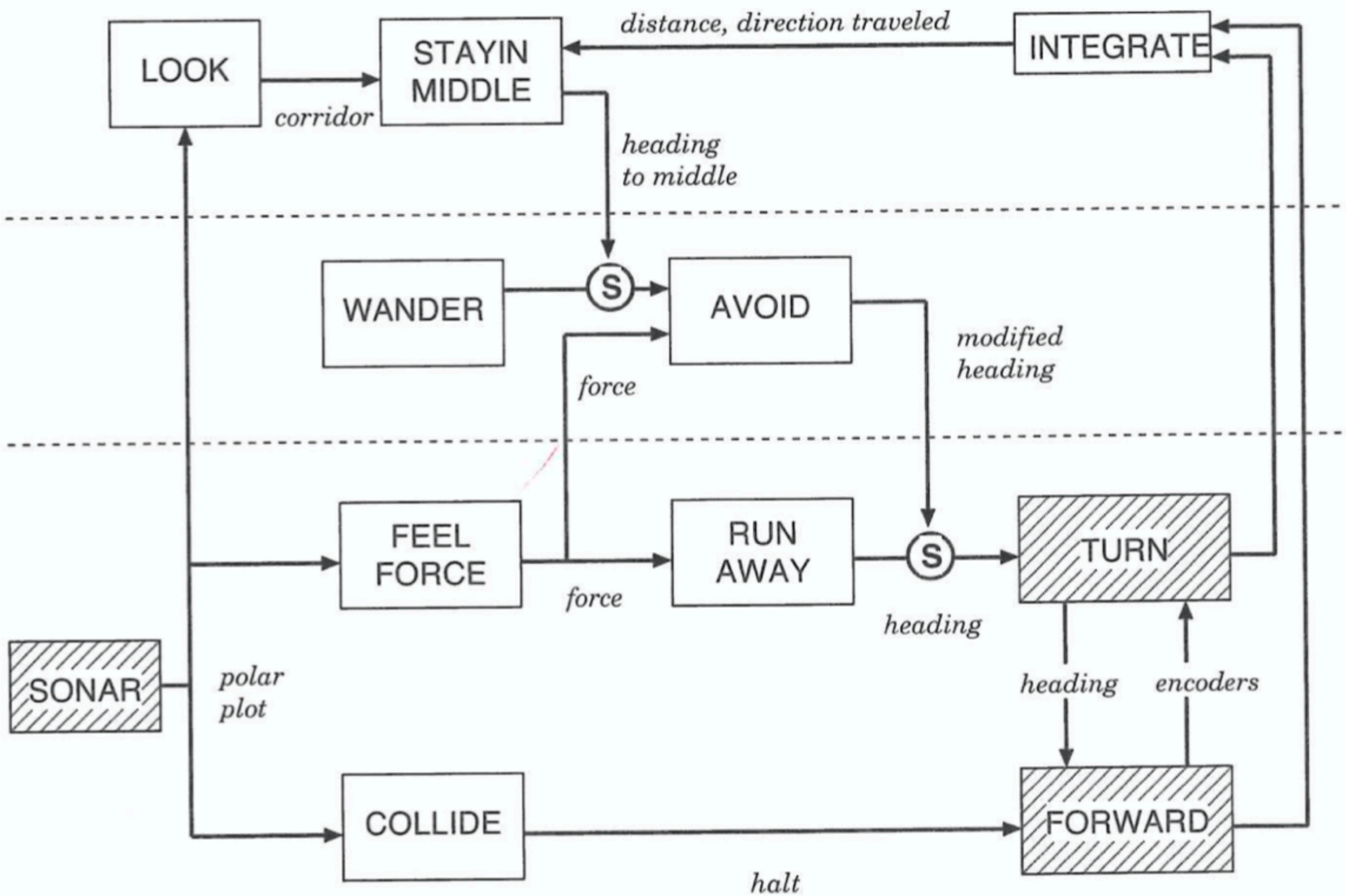


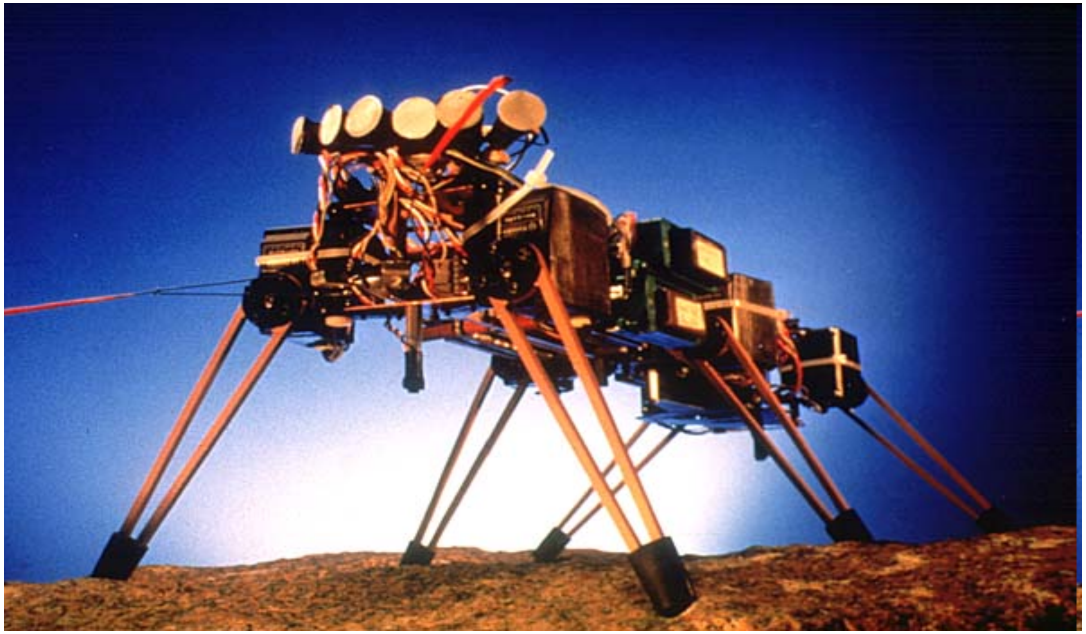
Figure 8.18 Level 2: follow corridors.

BEHAVIOURS AND ARBITRATION

- Behaviour
  - a piece of code executed as a thread is termed behaviour if it defines a control law for achieving and/or maintaining a given goal.
- Arbiter
  - a very specialized behaviour module monitoring robot operation which selects the most suitable behaviours to be activated depending on new acquired sensor.
- Arbitration
  - the implemented activity of the arbiter which results in the either spatial or temporal ordering of robot behaviours
- Robot Design
  - A specification of its sensor and effector devices
  - A definition of its basic behaviors along with its arbitration module

Behavior of Genghis:

- Standup
- Simple walk
- Force balancing
- Leg lifting
- Whiskers
- Pitch stabilization
- Prowling
- Steered prowling



EXAMPLE of decomposition of a complex behaviour

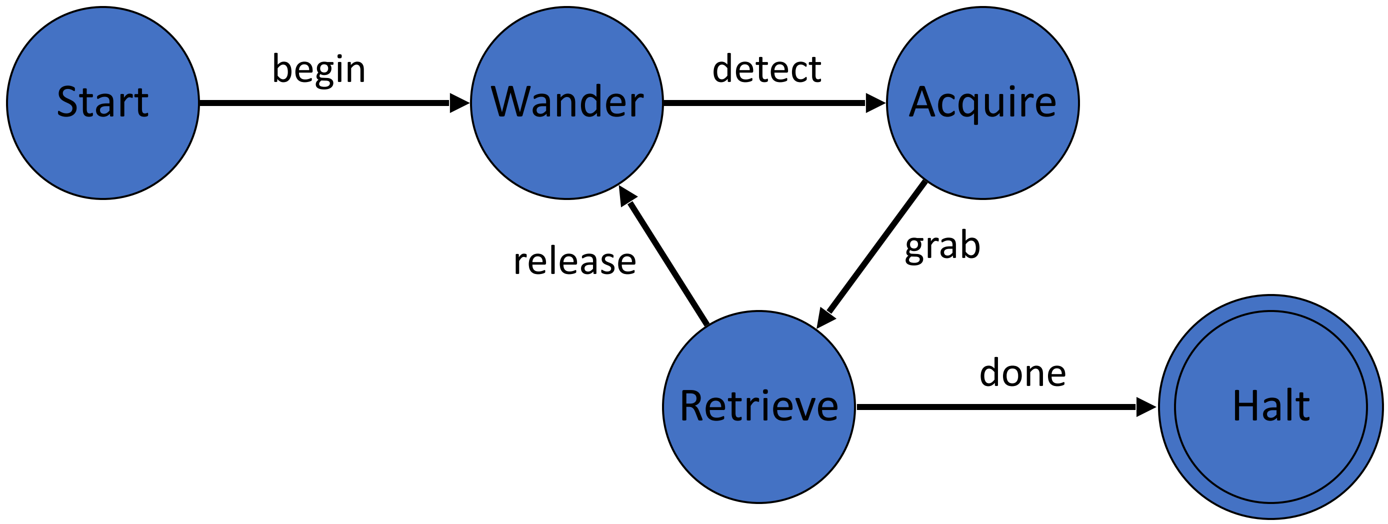
FORAGING:

- a robot moves from its base looking for objects that attract it
- when it finds an object, it moves toward it, pick it up, and returns to the base
- Repeats the sequence until all objects have been brought to the base

HIGH LEVEL BEHAVIORS REQUIRED:

- Wander: moving around the world searching for objects
- Acquire: move towards the object when it has been identified
- Retrieve: return the object to the base once it has been recovered

FSA DIAGRAM FOR THE FORAGING

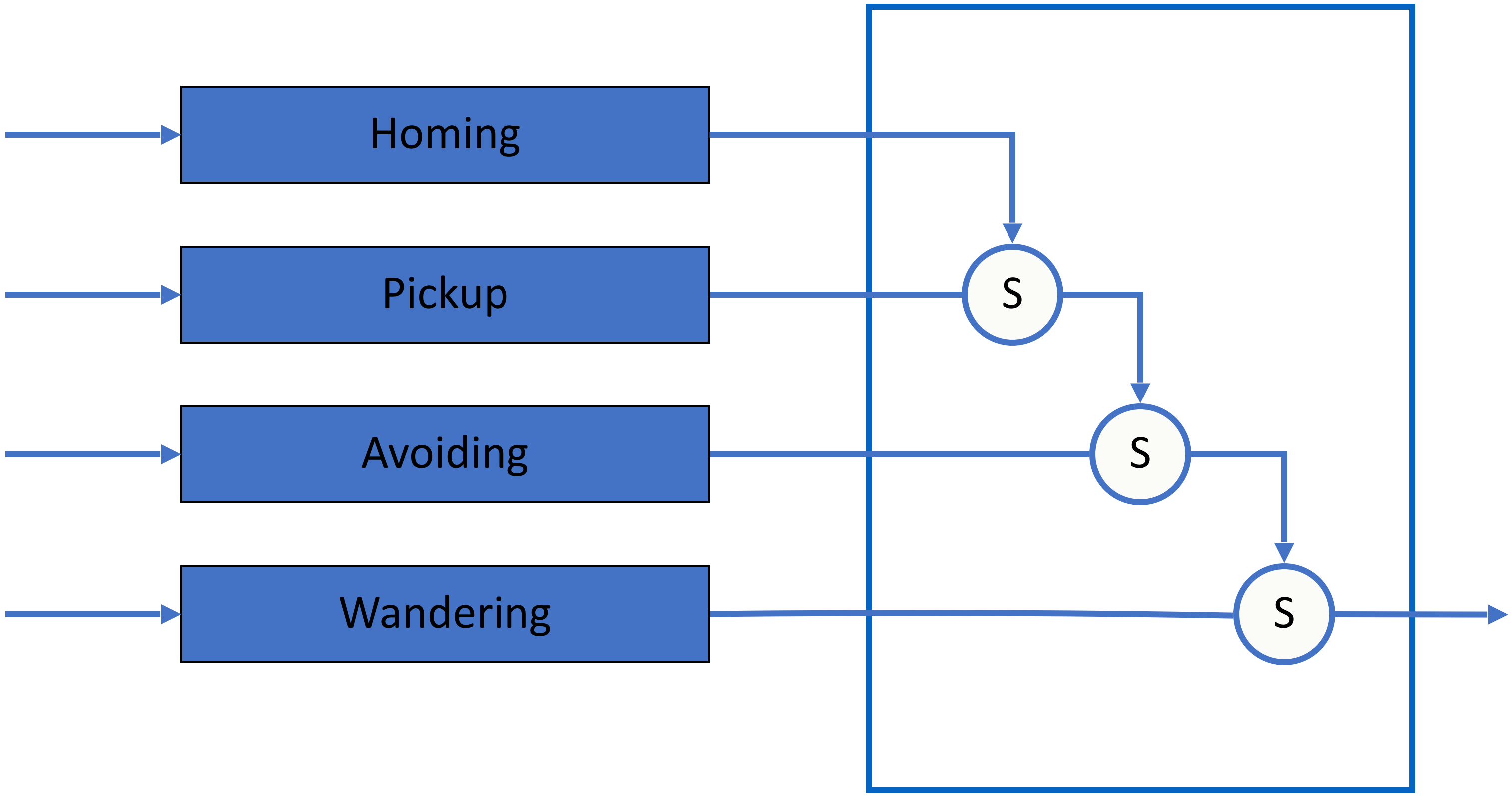


SOLUTION FOR THE FORAGING (1)

Behaviors involved:

- Wandering: move in a random direction for a while
- Avoiding:
  - turn right if the obstacle is on the left, then go ahead
  - turn left if the obstacle is on the right, then go ahead
  - after three tries, go back and turn around
  - If you have obstacles on both sides, turn around at a random angle and go back
- Pickup: turn to the object that attracts you and move on. If you have reached it, close the gripper
- Homing: turn towards the base and go on; if you have arrived, stop

SOLUTION FOR THE FORAGING (2)



INTERACTION OF CONCURRENT BEHAVIORS

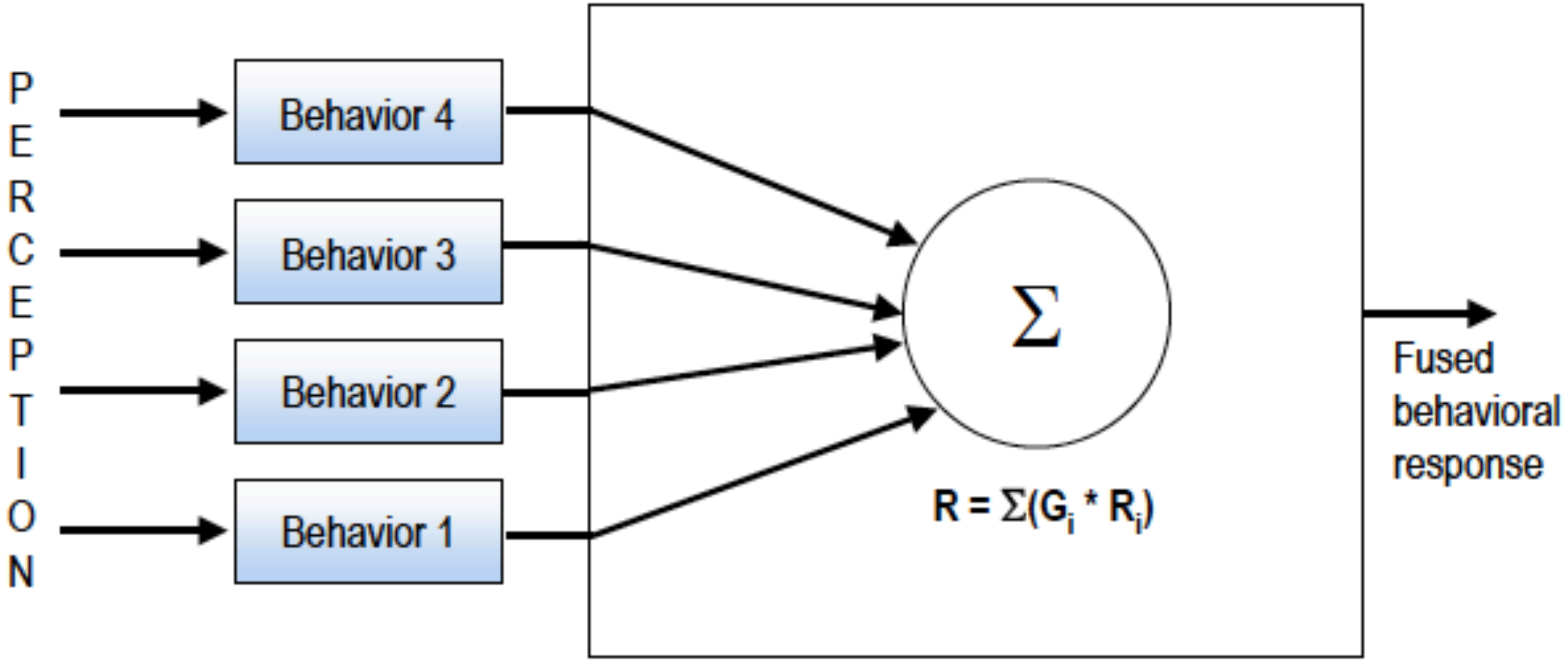
- Usually, behaviors follow a fixed sequence
- However, we can have multiple behaviors in certain situations
- How do behaviors interact?
  - Equilibrium
    - Behaviors balance each other out
    - Example: squirrel with food close to humans
  - Dominance of one/winner-take-all
  - Cancellation
    - Example: Male stickleback fish: if want to defend and fight (due to overlapping territories) end up building a nest

DEFINE THE COORDINATION STRATEGY

Two main strategies:

- Cooperative
  - It blends the outputs of different behaviors in some way so that they are consistent with the robot's final goal
- Competitive
  - Ex: pure arbitrage (only the output of one of the behaviors is selected
- Mixed
  - A combination of the two above

COOPERATIVE BEHAVIOR FUSION VIA VECTOR SUM



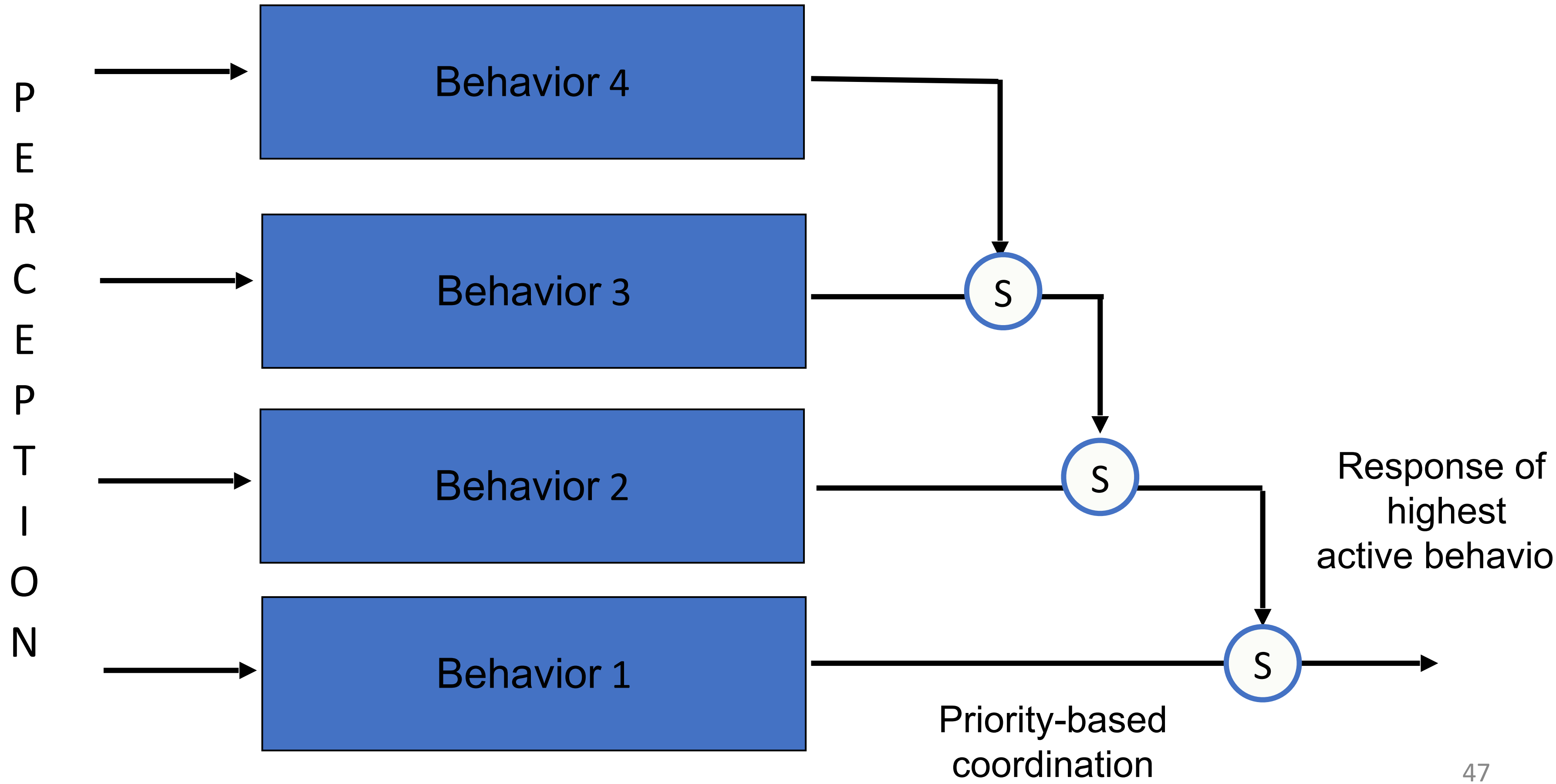
Behavioral fusion

- Positive example: corridor following
- Negative example: avoid obstale

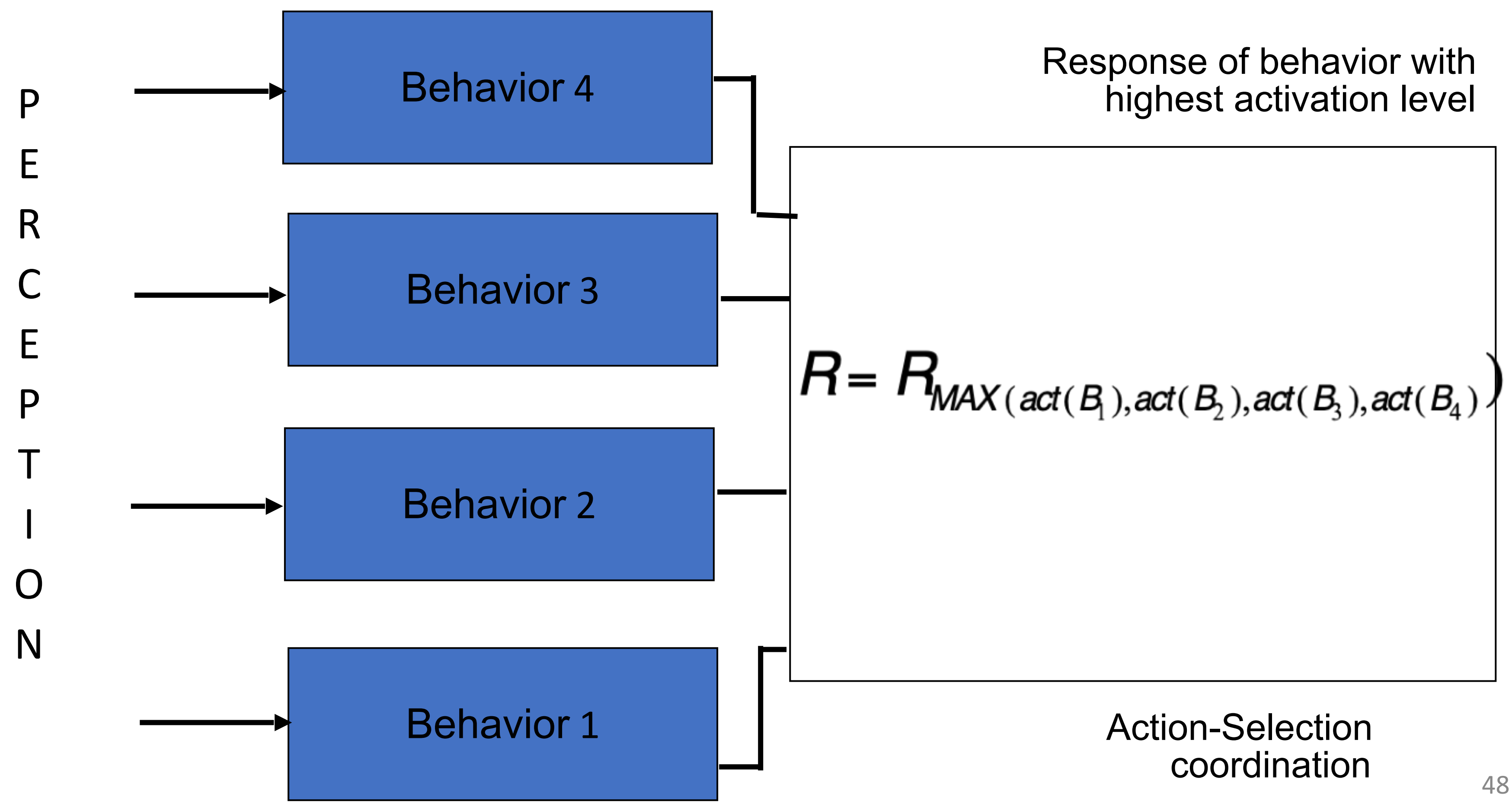
COMPETITIVE METHODS TO DEFINE BEHAVIOR

- They provide a way to coordinate responses in case of conflicts
- It can be seen as “the winner takes all”
- Arbitrage can be:
  - Fixed priorization
  - Action selection
  - Vote generation

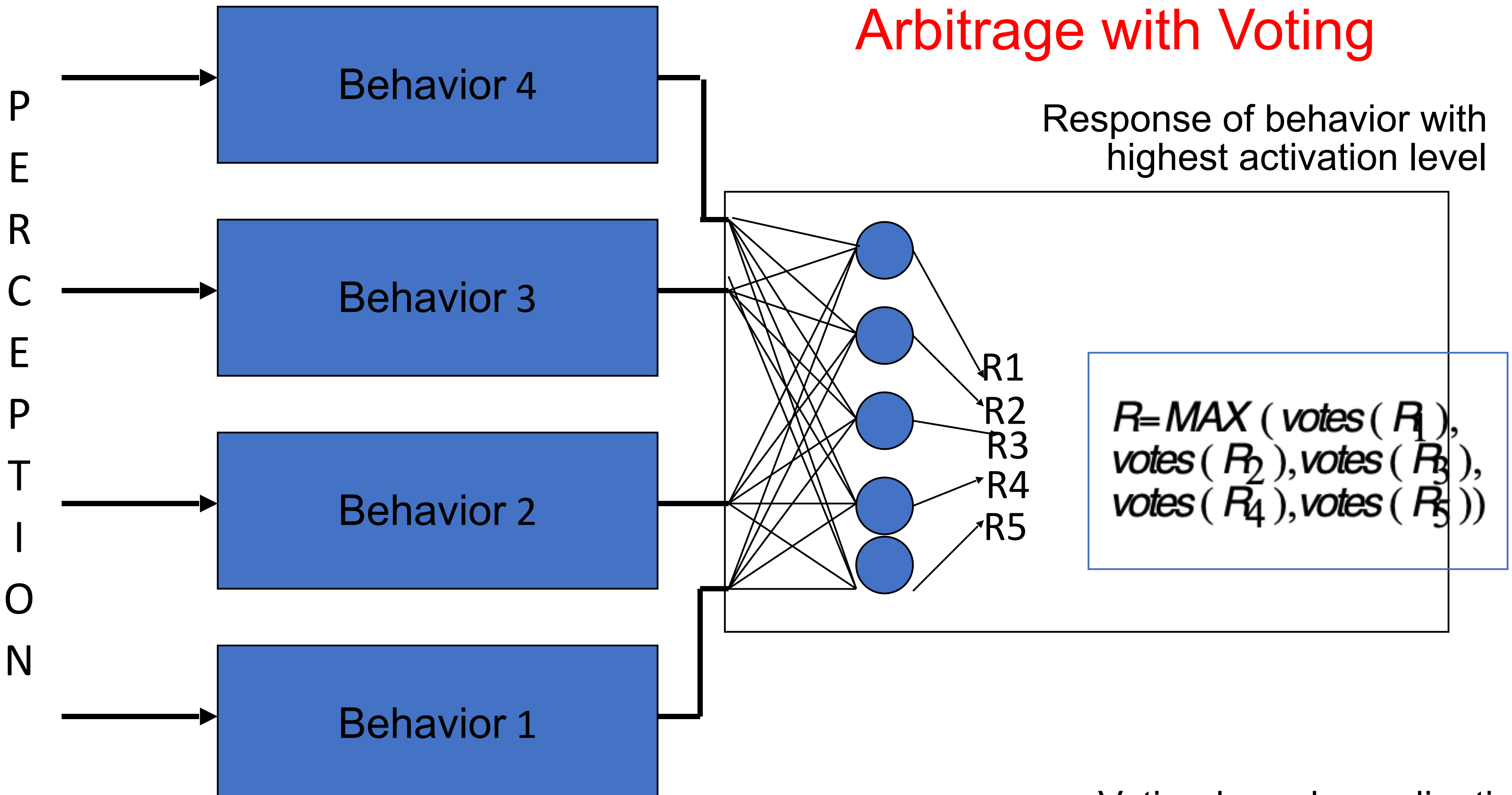
COMPETITIVE METHOD #1:
Arbitrage With Fixed Priorization



COMPETITIVE METHODS #2: Arbitrage with Action Selection



COMPETITIVE METHOD #3: Arbitrage with Voting



Voting-based coordination

BEHAVIORAL ROBOTICS ISSUES

- **Hidden Costs**
  - Behaviors plus some sort of ***coordination function*** to fuse outputs to effectors
- **Where it breaks**
  - "Fly at window" effects due to local scope
  - Poor choice of coordination functions (go left, go right = go in middle and hit obstacle)
- **Where it doesn't break but scares/annoys people**
  - often can't predict whether will go left or right to avoid (but only if doesn't matter)
  - Not optimal because sensor based (reactive) versus symbol/map based