based on Giuseppina Gini lectures on
BEHAVIOR BASED ROBOTICS

T. Ryan Fitz-Gibbon
BROOK'S SUBSUMPTIONARCHITECTURE

POLITECNICO MILANO 1863

Cognitive Robotics

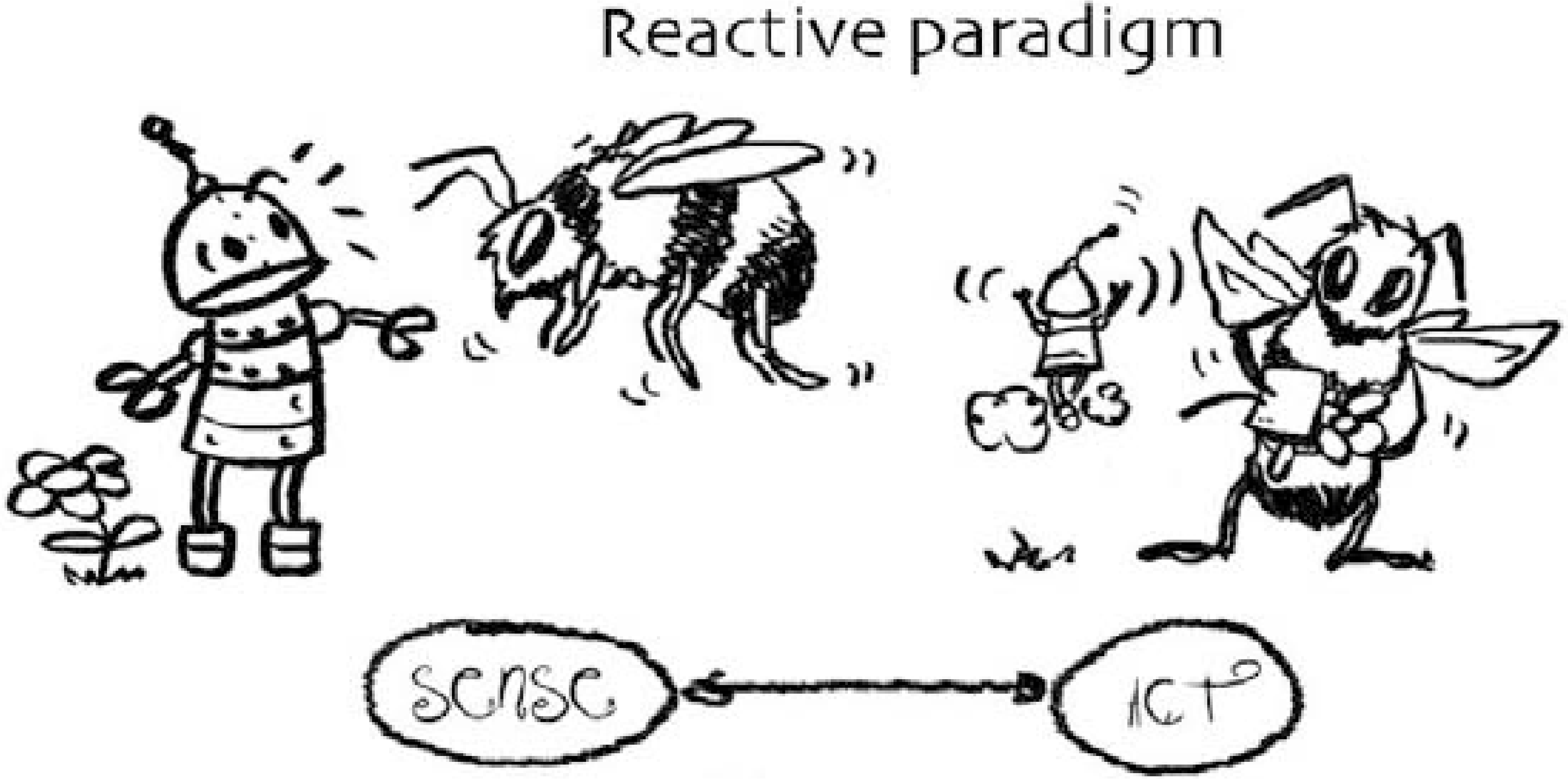
2017/2018

Reactive: Behavior Based Robotics

Matteo Matteucci
matteo.matteucci@polimi.it

Artificial Intelligence and Robotics Lab - Politecnico di Milano

Recall: «Don’t think, react!»



What is Intelligence?

A housefly is much simpler than most AI attempts, indeed it is unlikely it:

- Forms 3D surface descriptions of objects
- Reasons about the threat of a human with a fly swatter,
  in particular about the human’s beliefs, goals, or plans
- Makes analogies concerning the suitability for egg laying between dead pigs
- Constructs naïve physics theories of how to land on the ceiling

It is much more likely that a housefly:

- Has close connection of sensors to actuators
- Has pre-wired patterns of behavior
- Has simple navigation techniques
- Functions almost as a deterministic machine

And yet a housefly exhibit a more successful behavior in the real world than many AI attempts …

Other views on Intelligence and Robots

What is the mind – cognitive science answer

Where is the intelligence – behavioral approach

What is in the brain – neurosciences

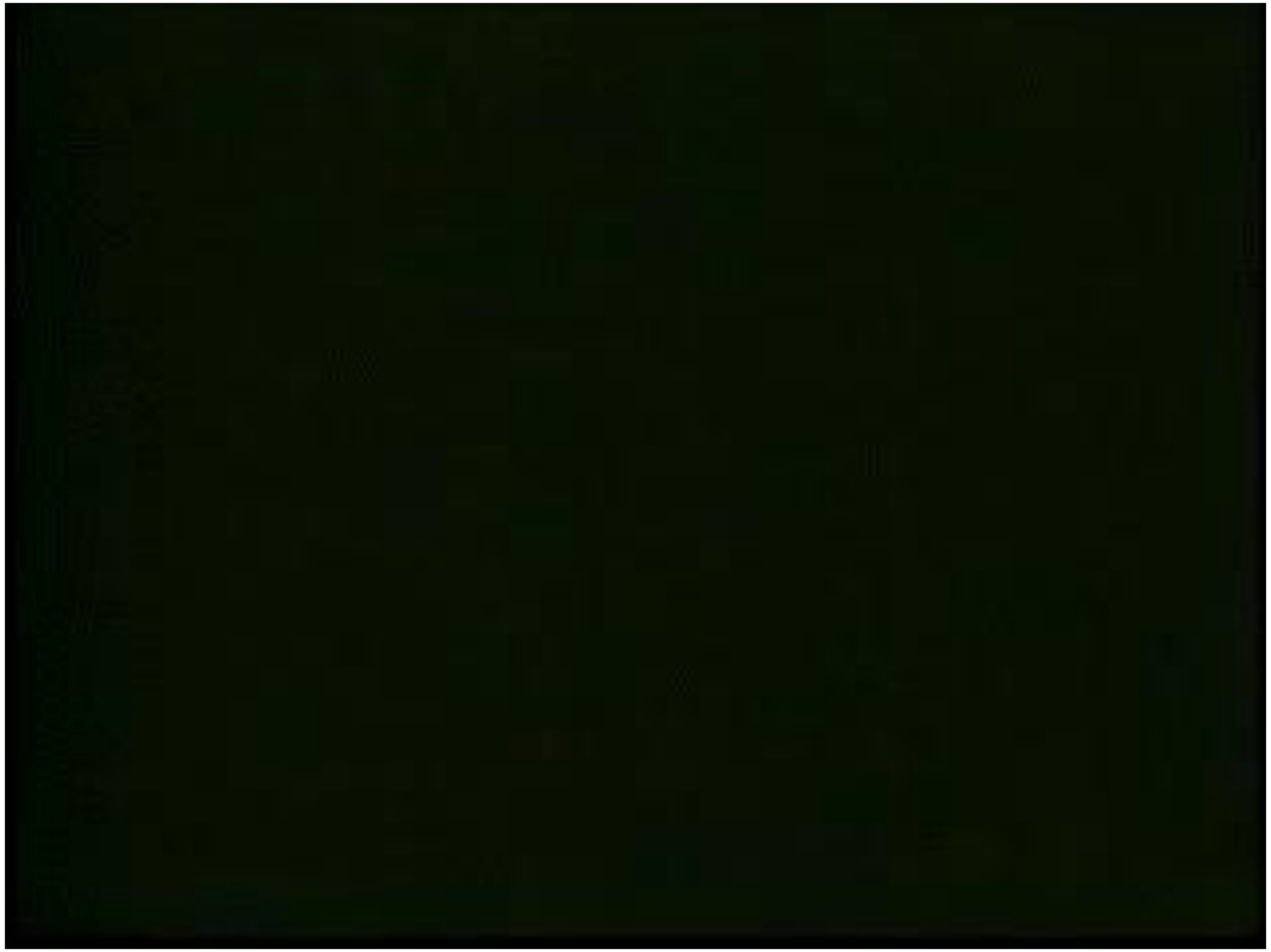
Alternative perspective in cognitive robotics

- 1950: Early experiments
- 1984: Behaviorism
- 1990: Subsymbolic

Parallel evolution with studies in Psycology

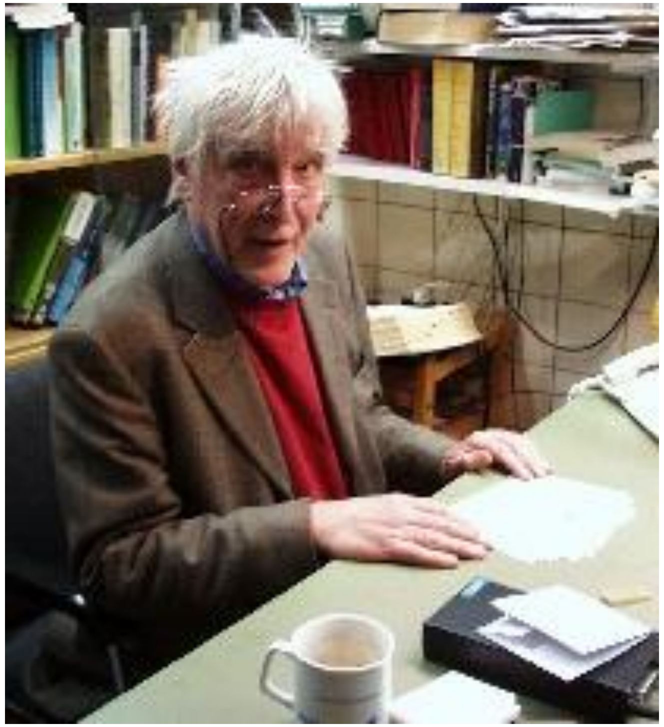
Early «robot» experiments

Grey Walter's tortoises ('50s): mechanical plausibility of animal tropism. (Tropism/taxis: animal movement directed by stimuli).



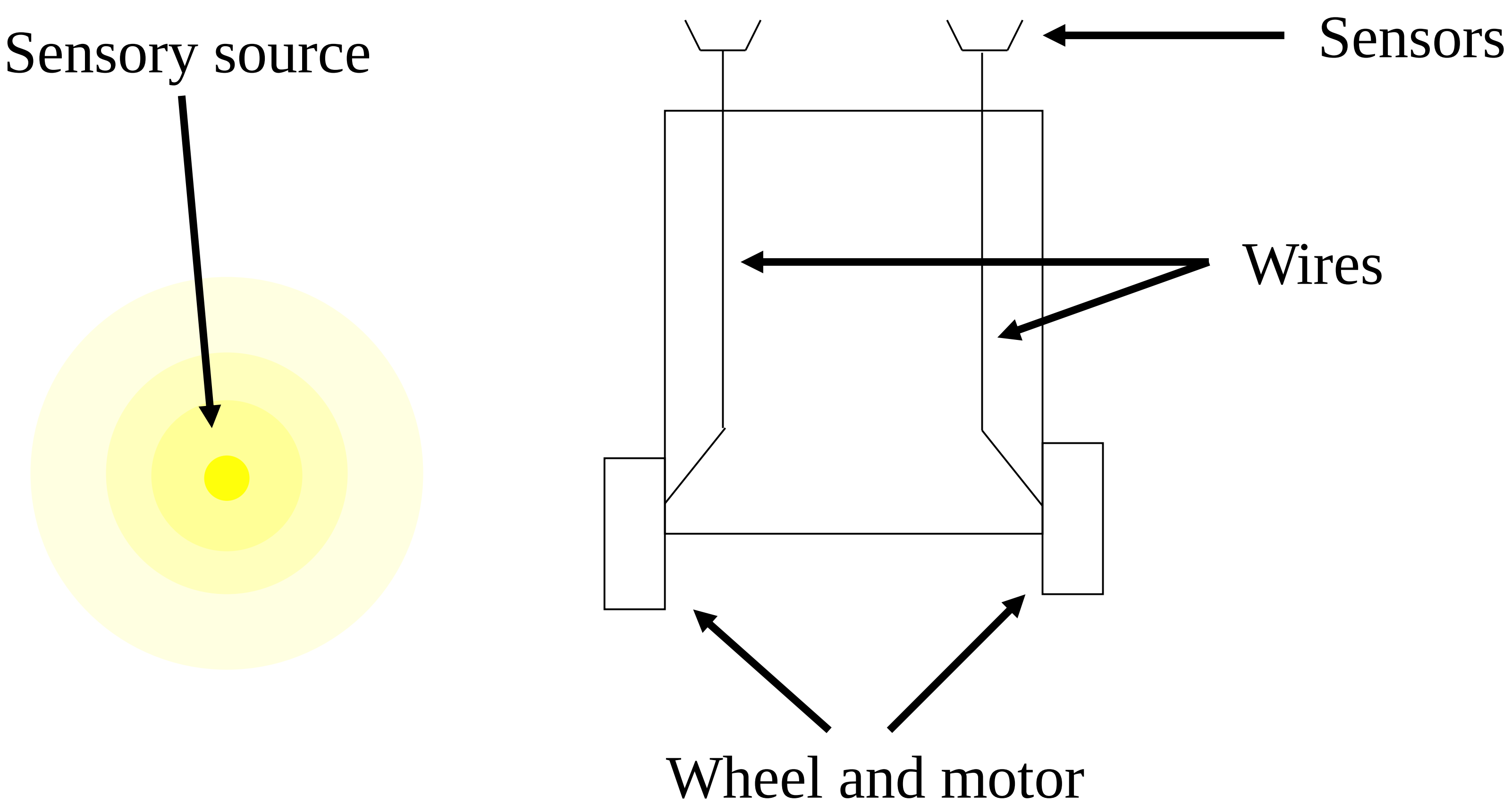
Braitenberg's Vehicles

- The book *"Vehicles: Experiments in Synthetic Psychology"* was published in 1984 by Valentino Braiteberg
- Neuro-psychologist interested in how primitive neural structures can give rise to complex behavior
- He developed a simple model of robots with sensors and motors to show how complex behavior can arise from simple mechanisms
- "Vehicles" where complex behavior emerges from combination of simple NNs encoding different taxes



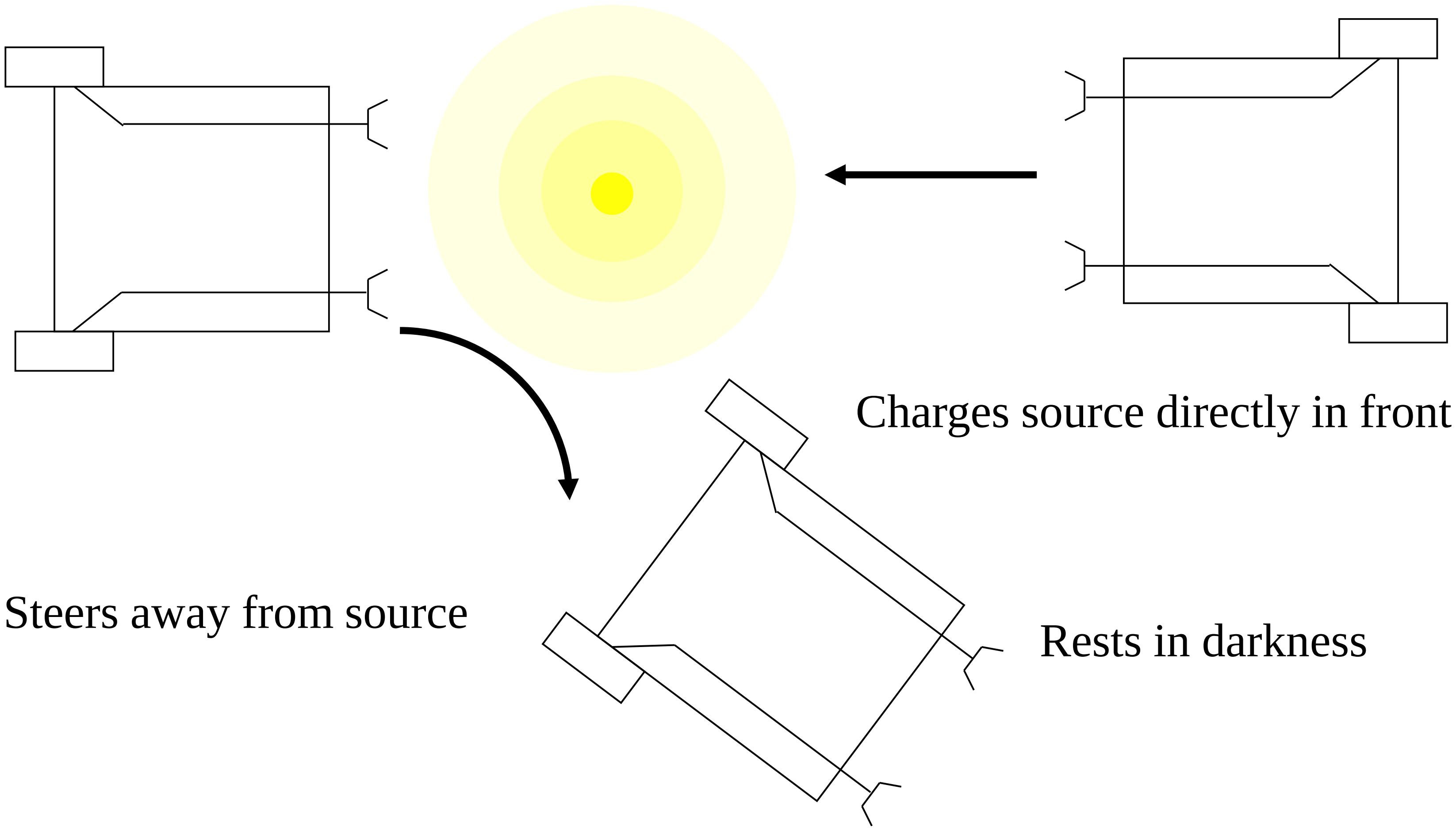
Dr. Braiteberg's homepage:
http://www.kyb.mpg.de/~braitenb

A Vehicle



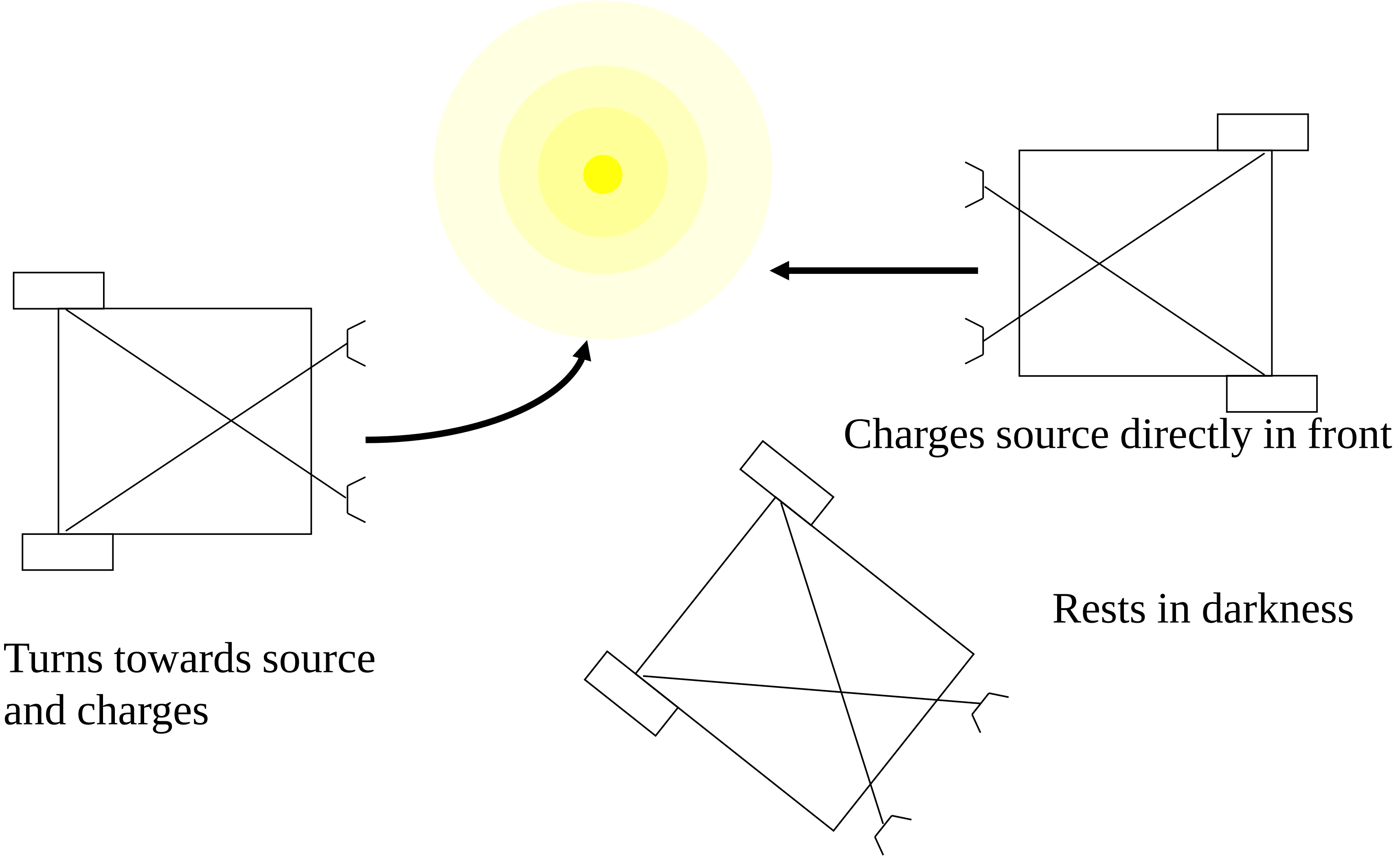
Vehicle 1: Coward

Sensors (light sensors) connected directly to motor on same side



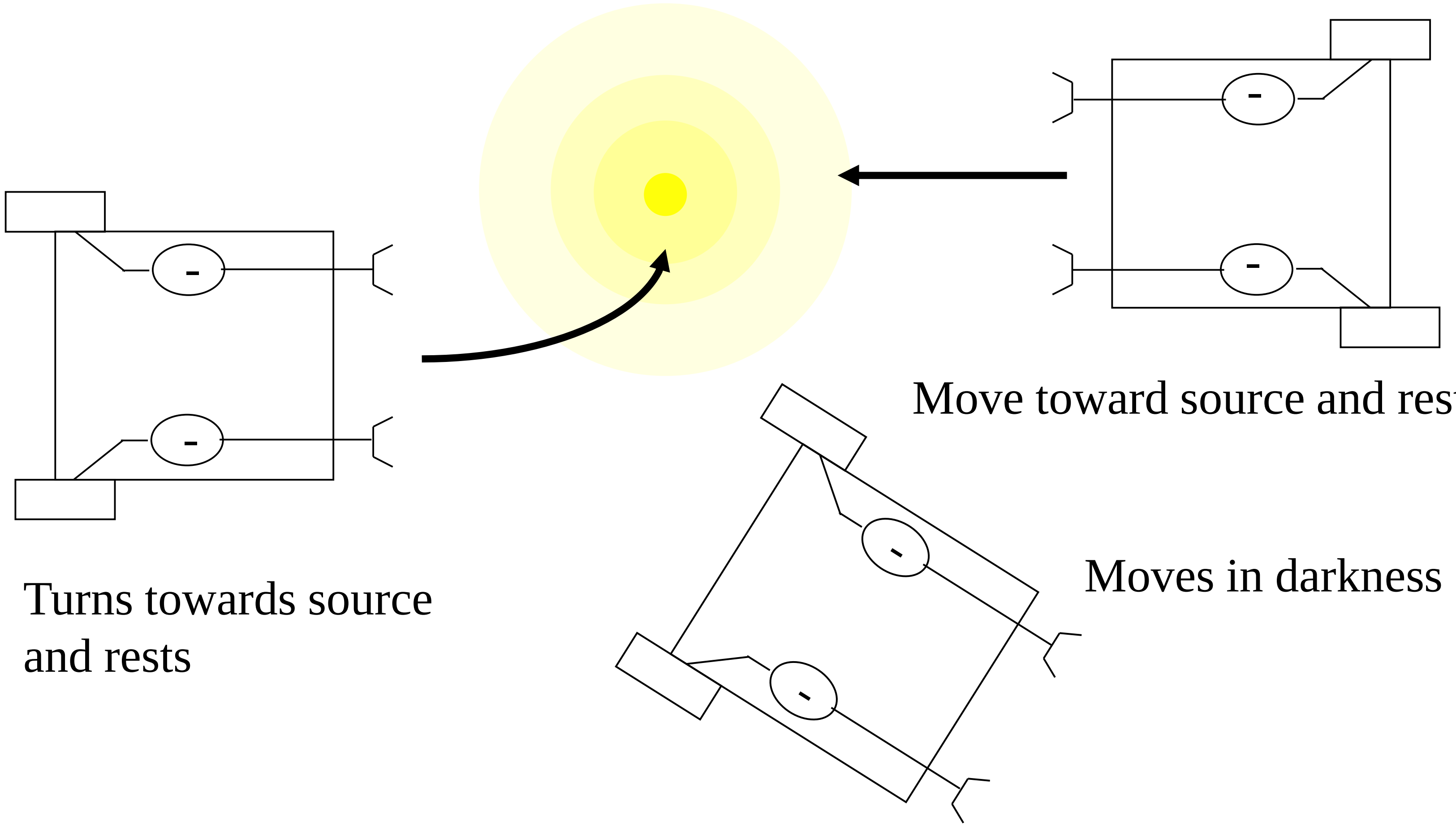
Vehicle 2: Aggressive

Sensors connected directly to motor on opposite side



Vehicle 3: Love

Sensors connected through inverter to same side



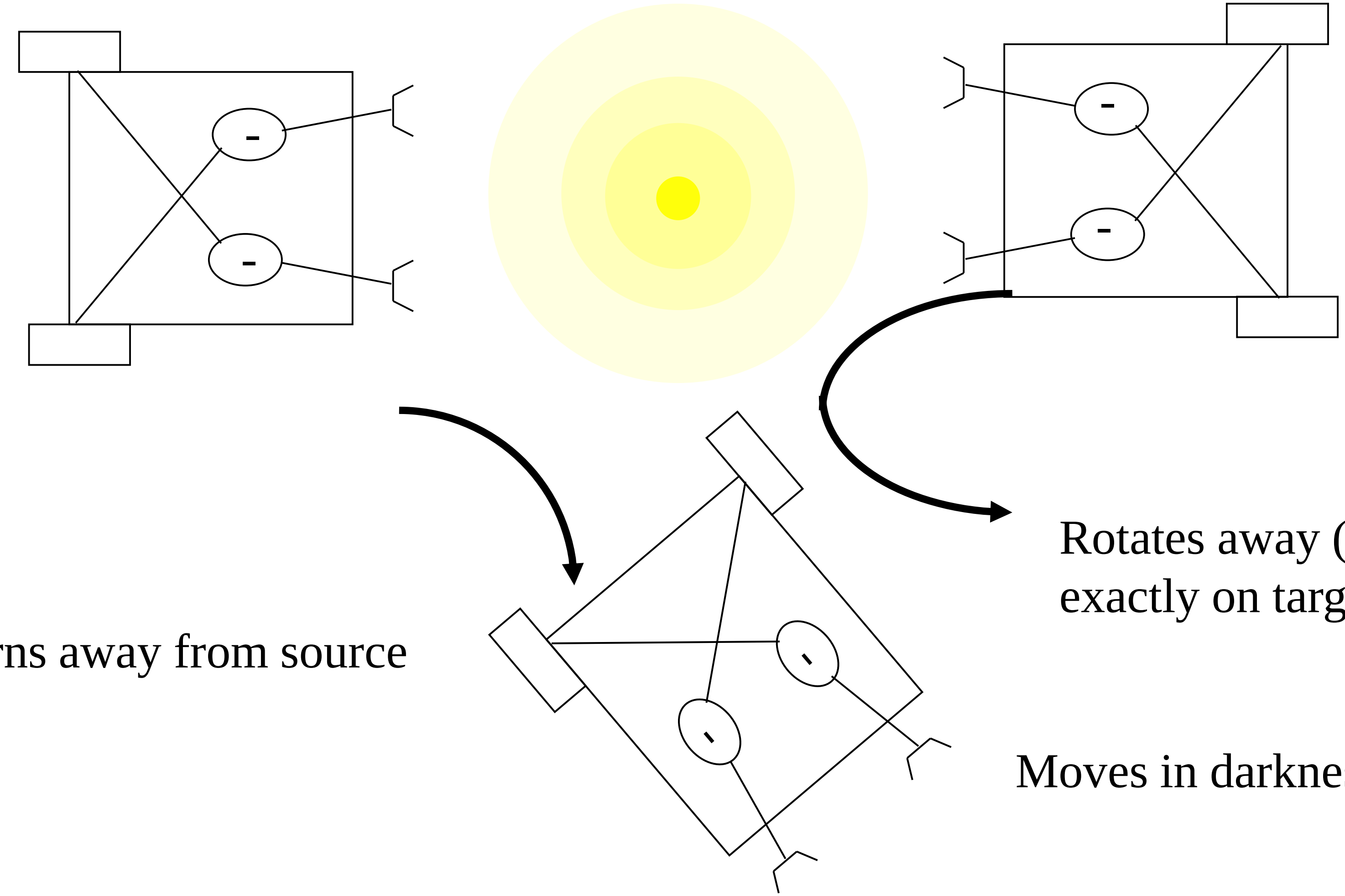
Move toward source and rest

Turns towards source
and rests

Moves in darkness

Vehicle 4: Explorer

Sensors connected through inverter to opposite side



Turns away from source

Rotates away (unless exactly on target)

Moves in darkness

Behaviorism

"Psychology should concern itself with
the observable behavior of people and animals,
not with unobservable events that take place
in their minds."

John B. Watson & B. F. Skinner

- Behaviorism emerged in the early twentieth century as a reaction to "mentalistic" psychology
- Nevertheless, in the second half of the 20th century, behaviorism was largely eclipsed as a result of the cognitive revolution

Behaviorism

"Psychology should concern itself with
the observable behavior of people and animals,
not with unobservable events that take place
in their minds."

John B. Watson & B. F. Skinner

- Psychology is the science of **behavior**, not the science of **mind**.
- Behavior can be described and explained without making ultimate reference to mental events or to internal psychological processes.
- The sources of behavior are external (they come from the environment), not internal (they do not come from the mind).

Behavior-based Robotics

An alternative approach to rethink how to organize intelligence born around 1984

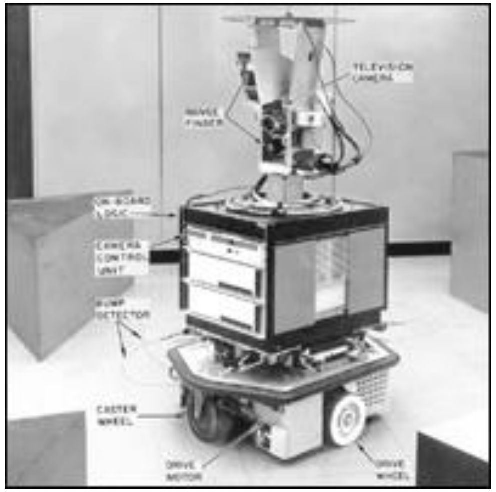
- Reactive to dynamic environment
- Operate on human time scales
- Robustness to uncertainty/unpredictability

Implemented simple systems with similar features

- Behavior language at MIT (Rodney Brooks)
- Schema at Georgia Tech (Ron Arkin)
- Fuzzy at SRI (Saffiotti, Ruspini, Konolige)
- Potential fields at Stanford (…)

A Change in Perspective

Reasoning		**Behave**
Where: brain		Where: organism
Artificial Intelligence		Artificial Life
Information processing		Senso-motor integration
Absolute coordinates		Agent-centered



Symbol grounding problem

"The symbol grounding problem is related to the problem of how words (symbols) get their meanings, and hence to the problem of what meaning itself really is. The problem of meaning is in turn related to the problem of consciousness, or how it is that mental states are meaningful."

Wikipedia

Classic AI, i.e., symbolic approaches, to Cognitive Robots have shown many unsolved (some would say insoluble) problems associated with symbol grounding:

- A purely symbolic system has no embodiment
- Symbol grounding not solved yet which means: no symbols, no symbolic relations, no automatic problem solution, no state space

http://cogprints.org/615/1/The_Symbol_Grounding_Problem.html

Properties of an Autonomous Robot

Situateness: the behavior of a creature depends on the environment in which it is embedded or situated, creatures don't deal with abstract descriptions, but with the "here" and "now" of their environment.

A situated automaton is a finite-state machine
whose inputs are provided by sensors connected to the environment,
and whose outputs are connected to effectors

Embodiment: An *embodied* creature is one which has a physical body and experiences the world, at least in part, directly through the influence of the world on that body.

Only an embodied agent is validated as one that can
deal with real world. Only through a physical grounding can
any internal symbolic system be given meaning.

Properties of an Autonomous Robot

Emergence: The intelligence of the system emerges from the system's interactions with the world and from sometimes indirect interactions between its components-- it is sometimes hard to point to one event or place within the system and say that is why some external action was manifested.

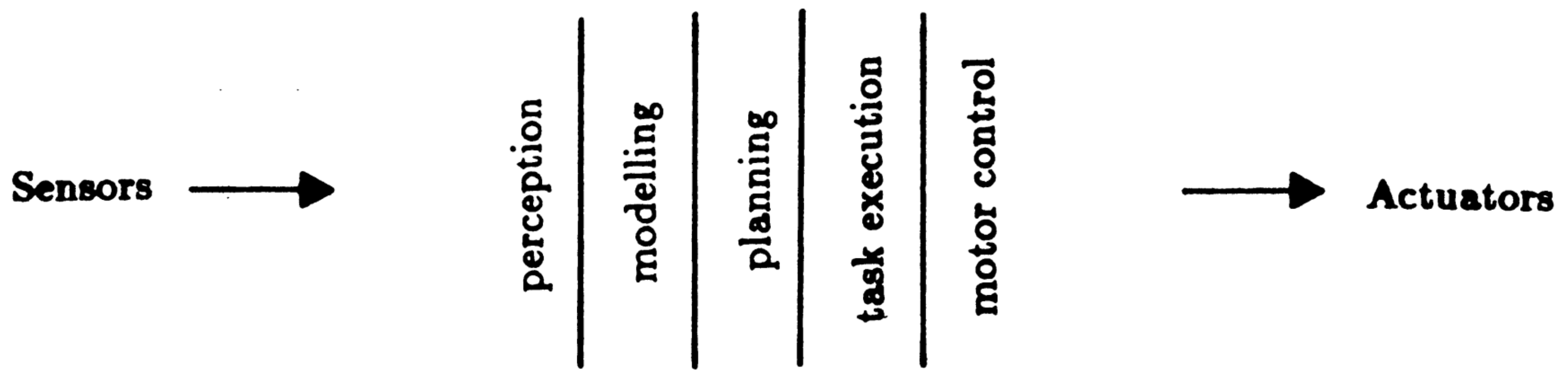
Intelligence: An autonomous (artificial) creature is one that is able to maintain a long term dynamic with its environment without intervention. Once an autonomous artificial creature is switched on, it does what is in its nature to do.

Intelligence is in the eye of the observer.

Horizontal vs. Vertical Decomposition

Shakey, yes but …

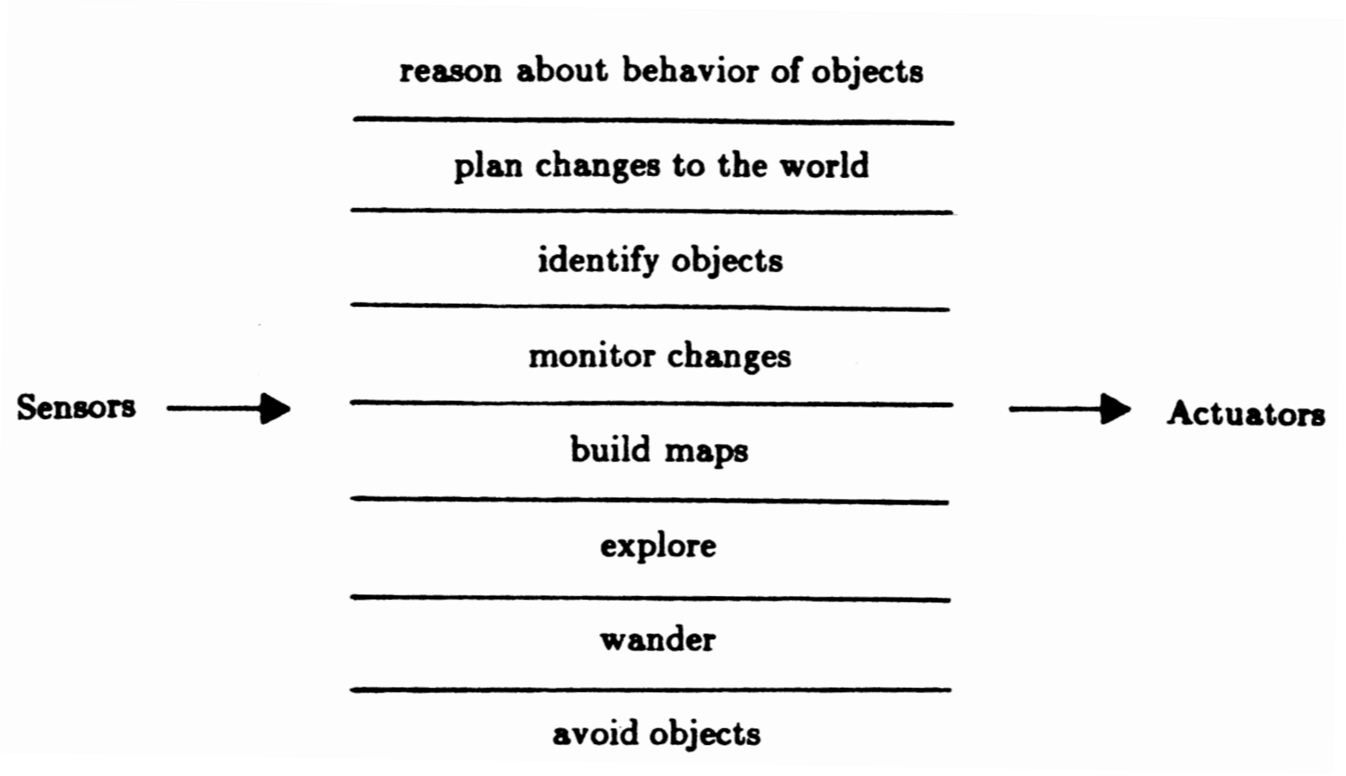
- Perception takes too long
- Perception is still not a solved problem
- Modeling/planning component assumes complete models available
- Overall system cannot respond in real-time
- Most robots built this way have failed



Horizontal vs. Vertical Decomposition

Decompose overall control system into a layered set of reactive behaviors

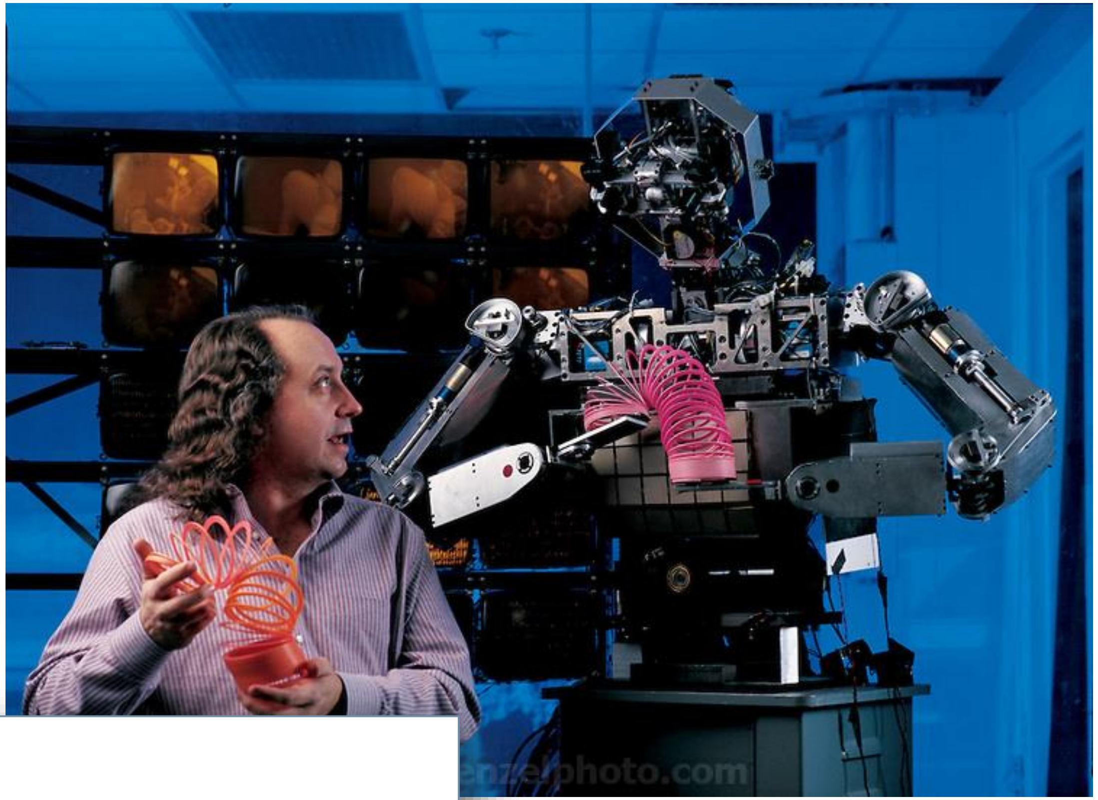
- Each behavior represents a complete mapping from sensors to motor commands
- Low-level behaviors (e.g., avoid) run in real-time since they use little computation
- High-level behaviors are invoked only when necessary
- Requires arbitration strategy to choose among (or combine) conflicting behaviors



Horizontal vs. Vertical Decomposition

Rodney Brooks has put forward three theses:

- Intelligent behavior can be generated without explicit representations of the kind that symbolic AI proposes
- Intelligent behavior can be generated without explicit abstract reasoning of the kind that symbolic AI proposes
- Intelligence is an emergent property of certain complex systems



Intelligence without representation*

Rodney A. Brooks

MIT Artificial Intelligence Laboratory, 545 Technology Square, Rm. 836, Cambridge, MA 02139, USA

Received September 1987

Brooks, R.A., Intelligence without representation, Artificial Intelligence 47 (1991), 139–159.

The Subsumption Architecture

Complex behavior needs not necessarily be a product of a complex control system

- Absolute coordinate systems are a source of error while Relational maps are more useful to a mobile robot
- Robots should be autonomous and self-sustaining, and able to function when one or more of its sensors fails recovery should be quick

To illustrate his ideas, Brooks built some robots based on his subsumption architecture:

- A subsumption architecture is a hierarchy of task-accomplishing behaviors
- Each behavior is a rather simple rule-like structure
- Each behavior ‘competes’ with others to exercise control over the agent
- Lower layers represent more primitive kinds of behavior (such as avoiding obstacles), and have precedence over layers further up the hierarchy

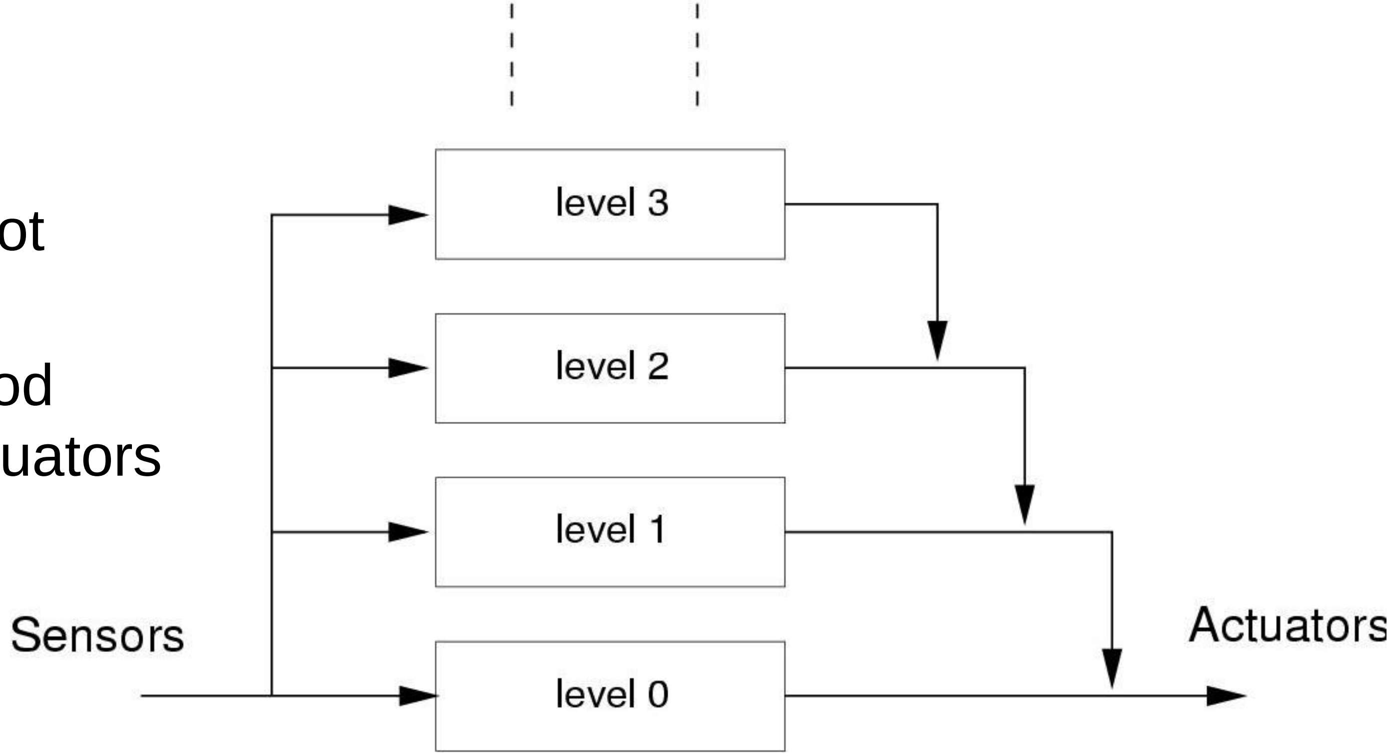
Subsumption Architecture Principles

A complex system has precursor in a simpler one

- The offspring contains the same mechanisms of the father with something more
- The offspring sussumes the father mechanisms

The Subsumption Architecture is:

- A layering methodology for robot control systems
- A parallel and distributed method for connecting sensors and actuators



Each level sussumes as a subsystem the previous level

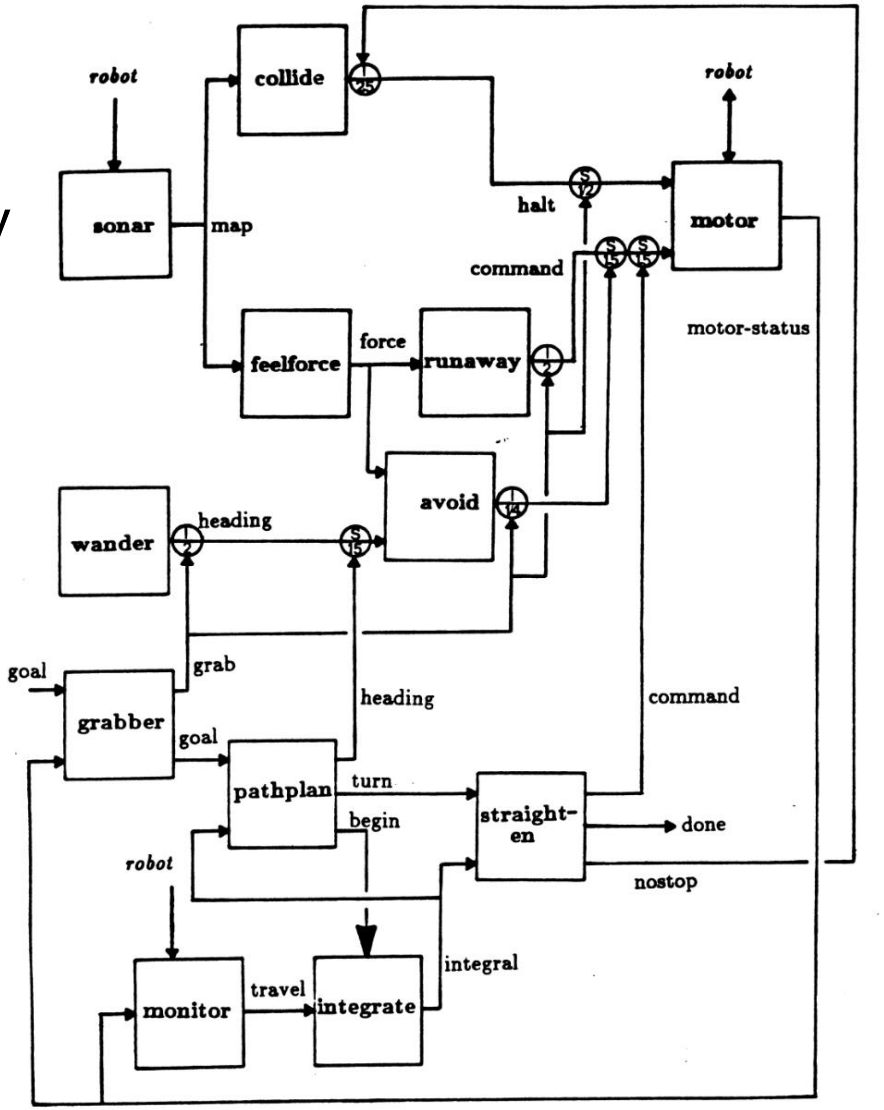
Computation through Augmented Finite State Machines

Computation is organized as an asynchronous network of active computational elements (AFSM) with fixed topology

- Messages have no implicit semantics, just few bits
- Message meanings are dependent on the dynamics designed into both the sender and receiver
- Sensors and actuators are connected through asynchronous two-sided buffers.
- Only internal timers

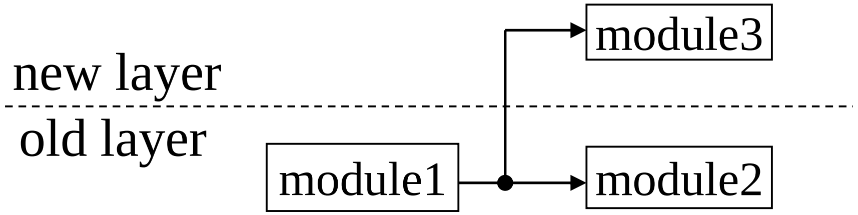
The system is broken down into parallel behaviors

- Each behavior has direct access to sensor readings and can control the robot's motors directly

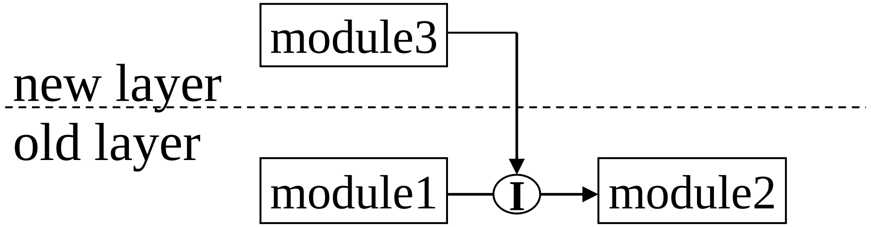


Communication Among AFSMs

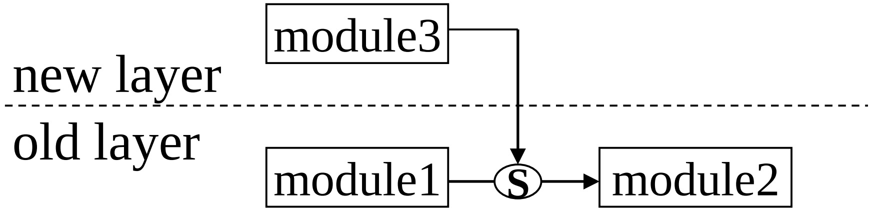
A new layer **monitors** messages in the previous layer using the same connection



It can **inhibit** the communication for a given time (e.g., 40 ms)



It can **substitute** the message for a limited time (40 ms)

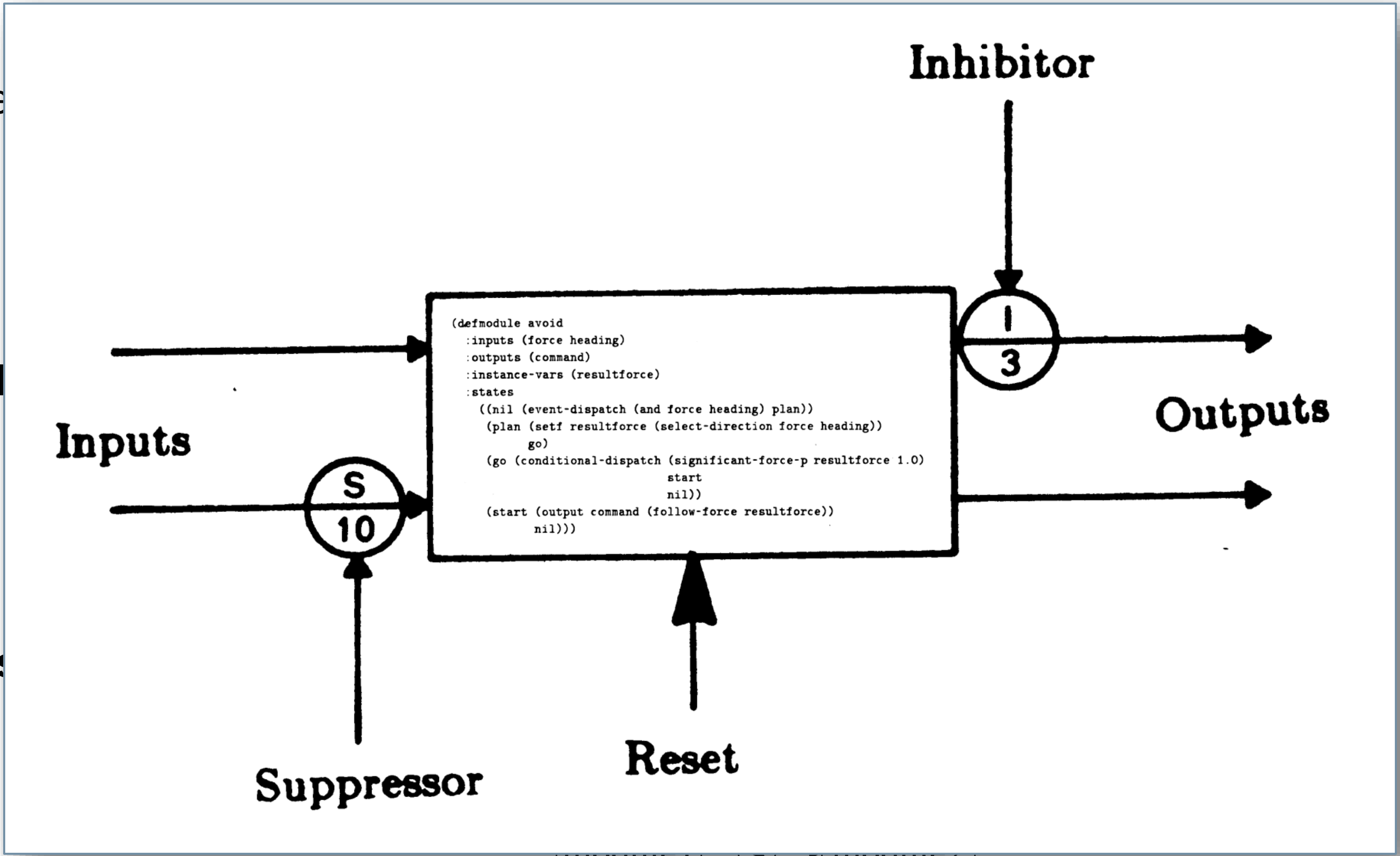


Communication Among AFSMs

A new laye

It can **inhil**

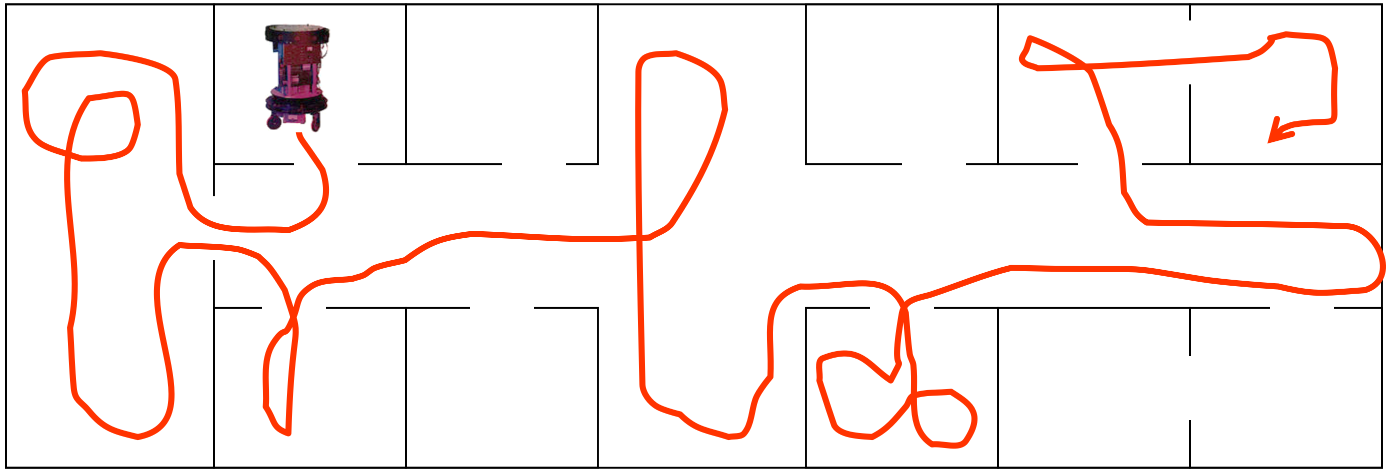
It can **subs**

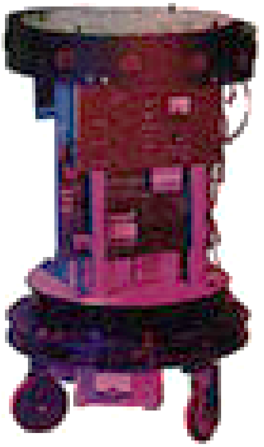


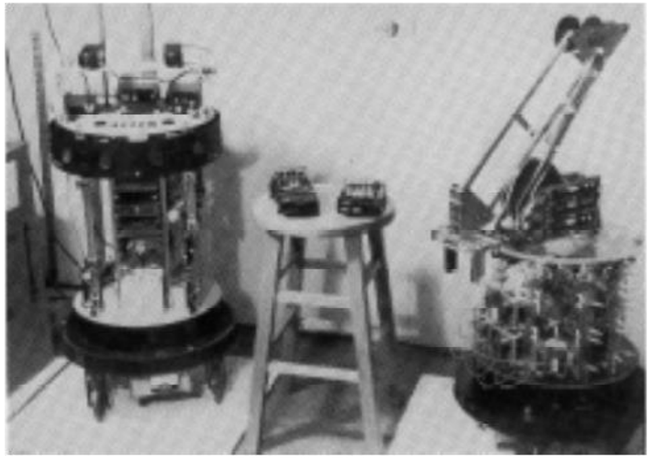
module1 S module2

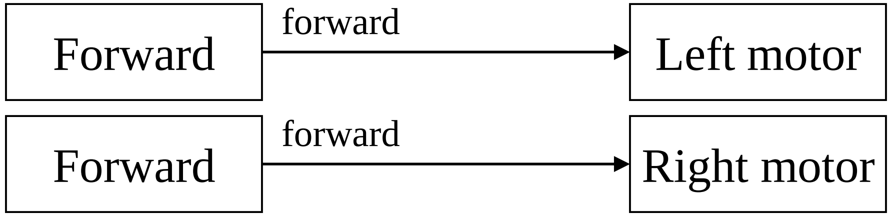
Example: Allen

1. *The robot moves only forward*



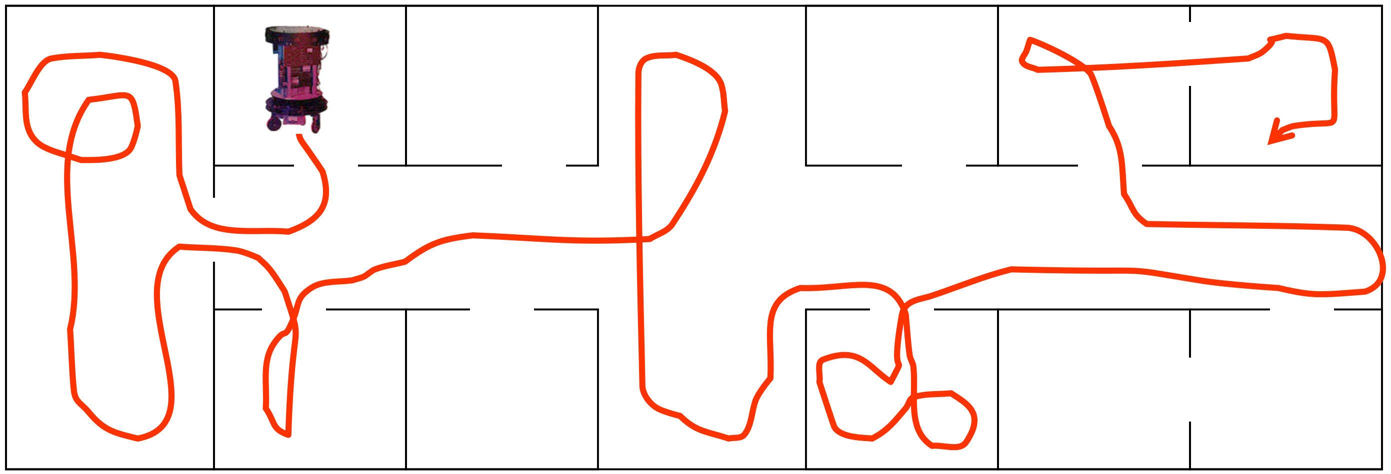


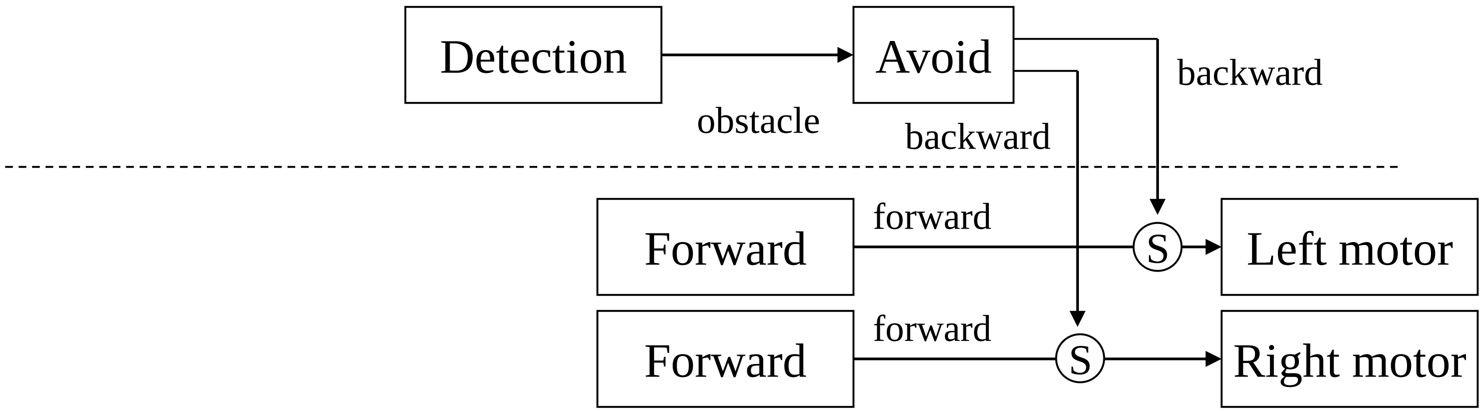




Example: Allen

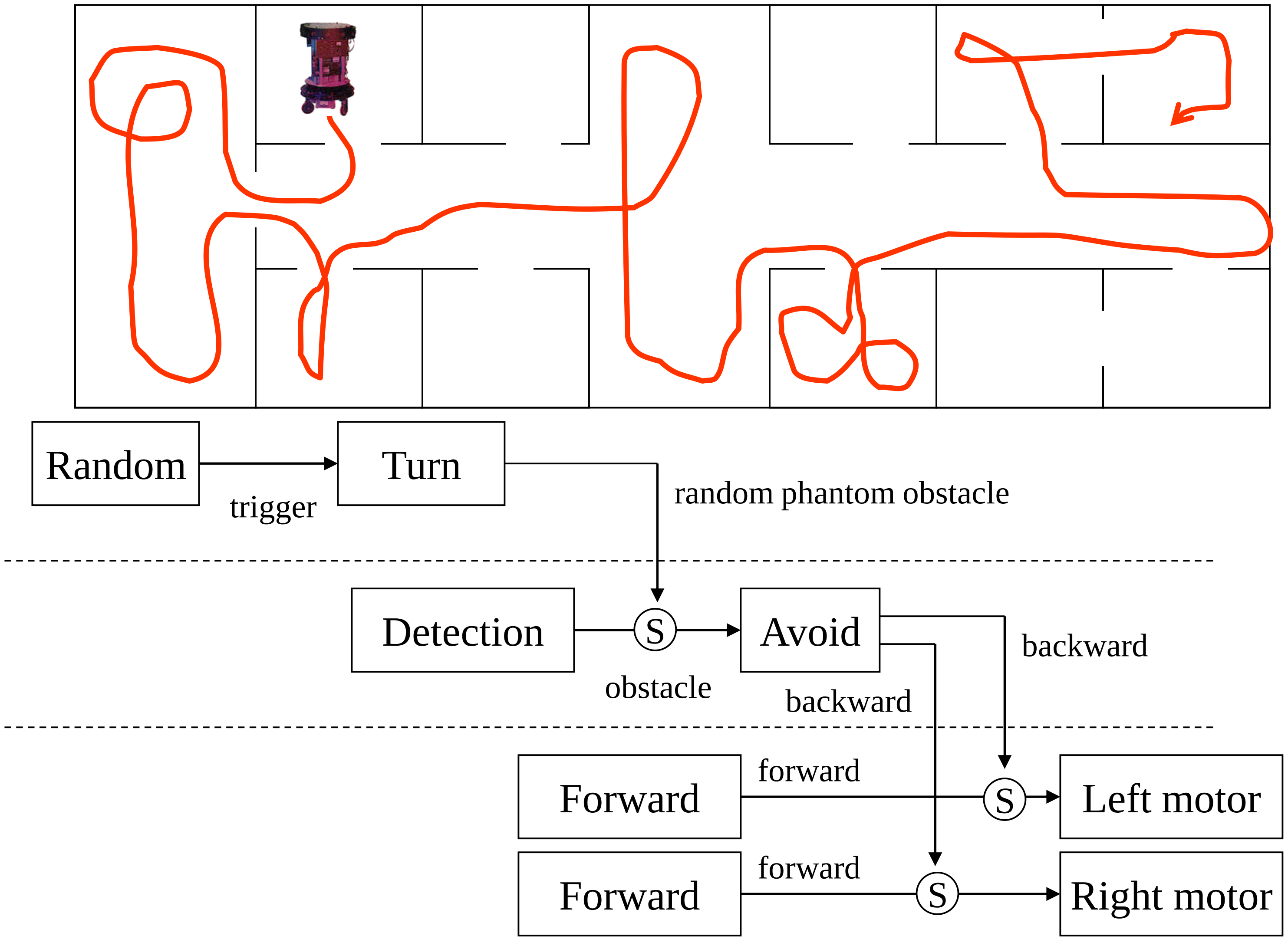
1. *The robot moves only forward*
2. *A new layer to look for obstacles. If obstacle detected, one wheel backwards to change direction.*





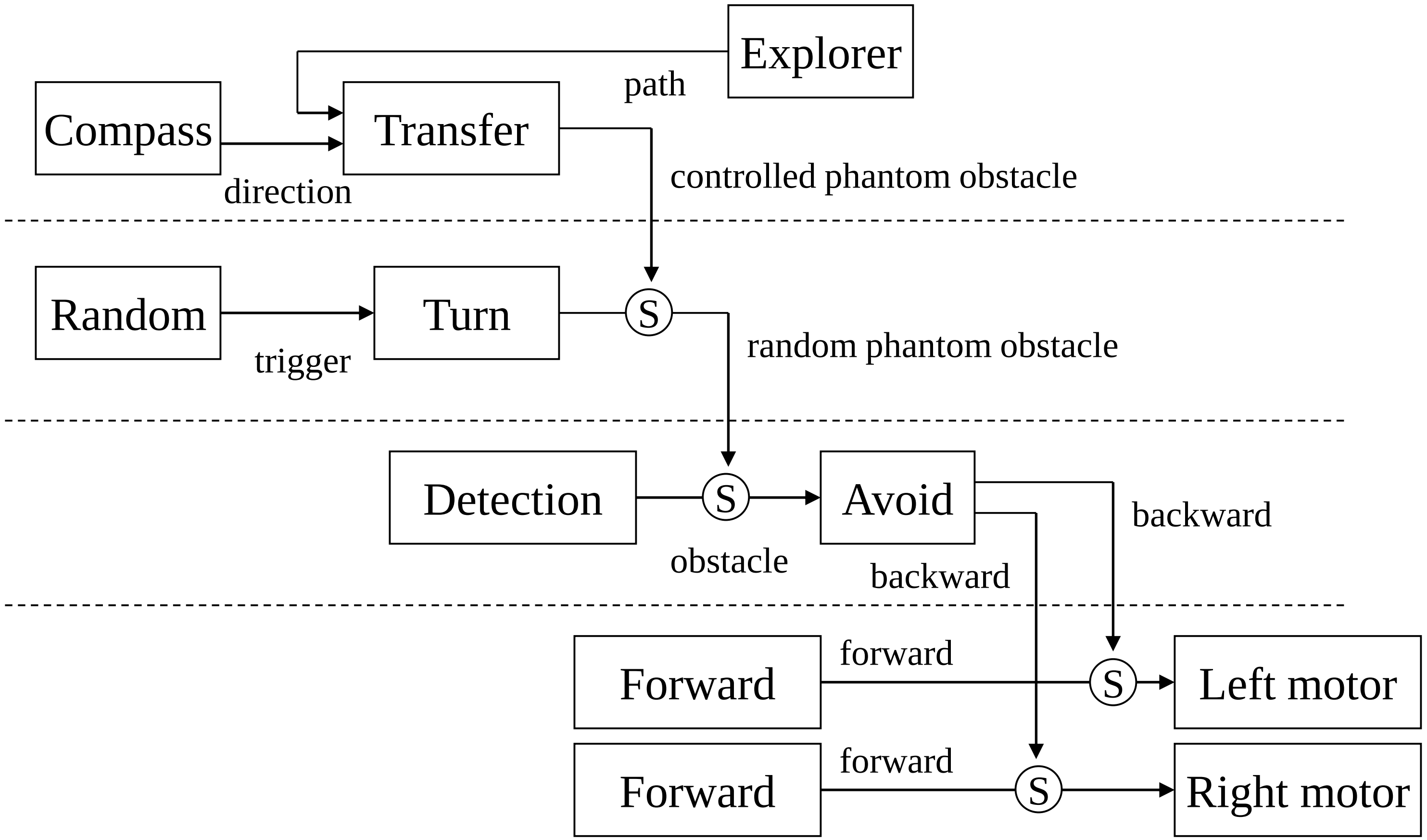
Example: Allen

1. *The robot moves only forward*
2. *A new layer to look for obstacles. If obstacle detected, one wheel backwards to change direction.*
3. *To avoid the robot continues to move in a small region, a new level will introduce random "phantoms" to make the robot change direction*



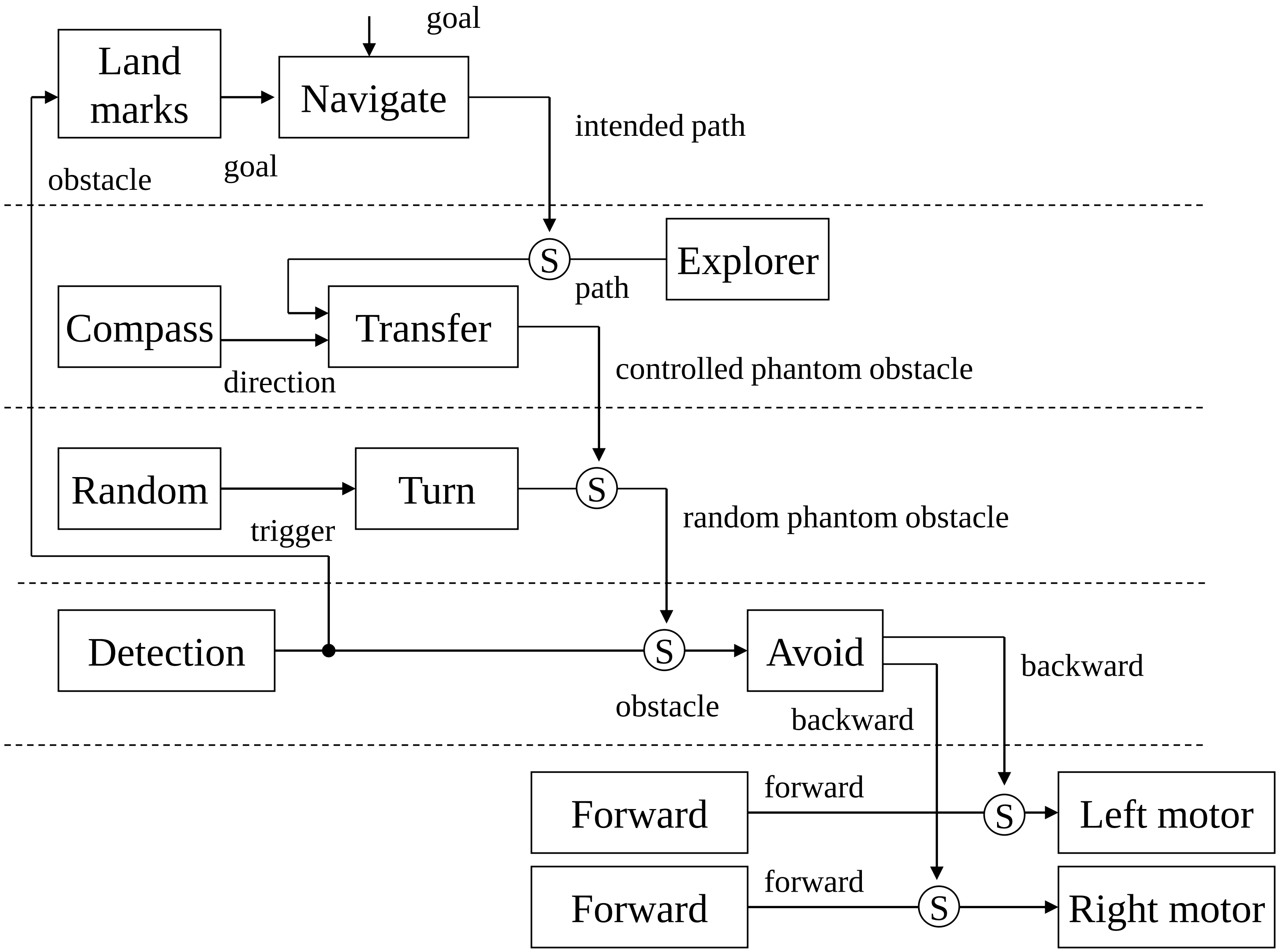
Example: Allen

1. *The robot moves only forward*
2. *A new layer to look for obstacles. If obstacle detected, one wheel backwards to change direction.*
3. *To avoid the robot continues to move in a small region, a new level will introduce random “phantoms” to make the robot change direction*
4. *To follow a wanted direction, we control the phantom obstacle*



Example: Allen

1. *The robot moves only forward*
2. *A new layer to look for obstacles. If obstacle detected, one wheel backwards to change direction.*
3. *To avoid the robot continues to move in a small region, a new level will introduce random "phantoms" to make the robot change direction*
4. *To follow a wanted direction, we control the phantom obstacle*
5. *Find landmarks. The user sends a direction, the robot moves toward the landmark*



Behavior Design

The world really is a rather good model of itself

No internal model of the real world

- No free communication, No shared memory
- Use real world as the model (Very accurate, Never out of date,
  No computation needed to keep model up to date, Real world used for sub-system communication because sub-systems just sense the real world)

Nevertheless behavior design is more an art than a science

- In what situation does the behavior apply?
- What is the result of the behavior?
- What robustness can we expect?
- What is the real scalability of the approach?

The emergent behavior is difficult to predict!

Example: herbert

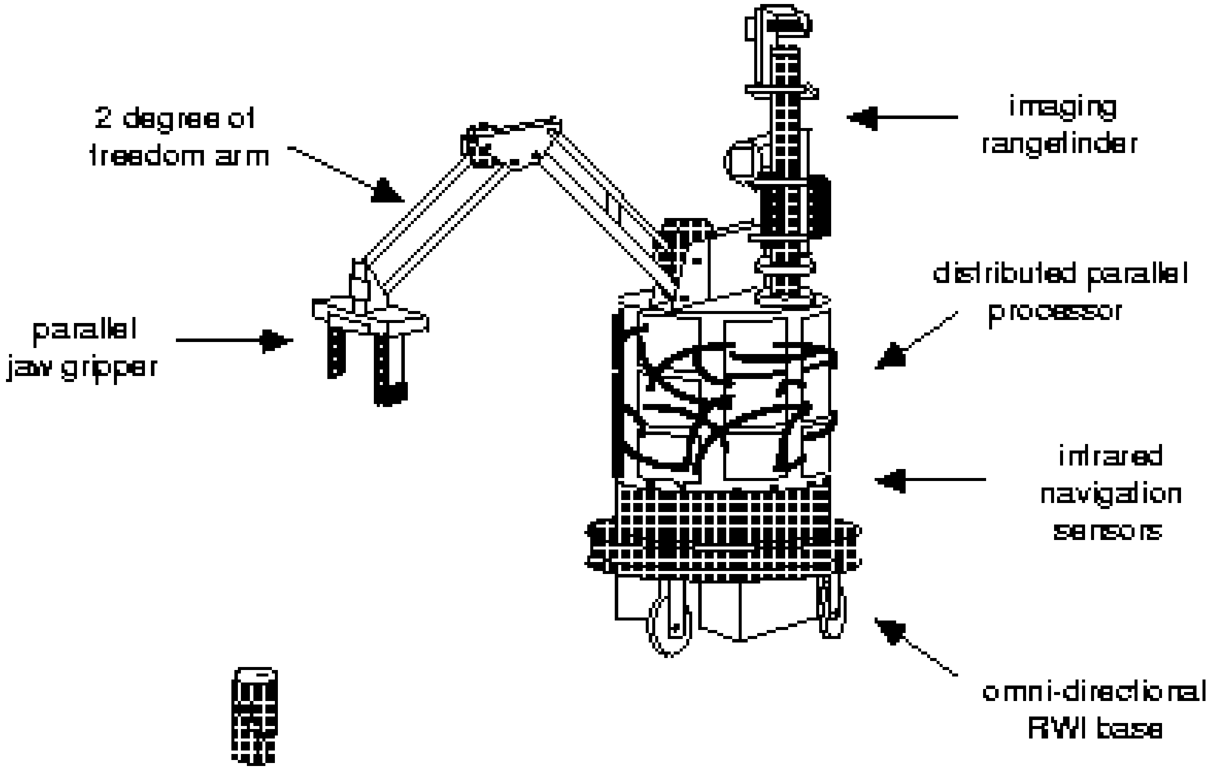
MIT AI Lab in 1988

Primary Goal

- To wander in the office following walls and avoiding obstacles
- Look for soda-like objects, pick it up and drop it at a base location

Sensors

- vision and a laser striper to find soda cans, and sonars and IRs to wander around safely.



Example: Herbert Grasping Behavior

Case 1: Can already between the fingers

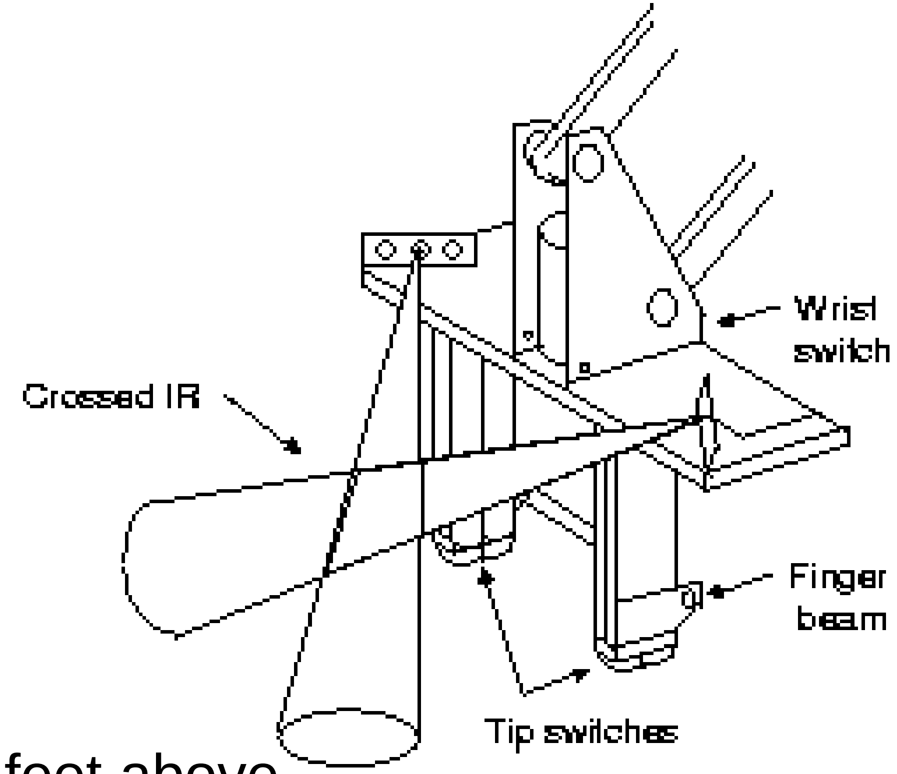
- Identified by the finger beam
- It grips everything that can be placed between the fingers

Case 2: Can placed in front of the hand

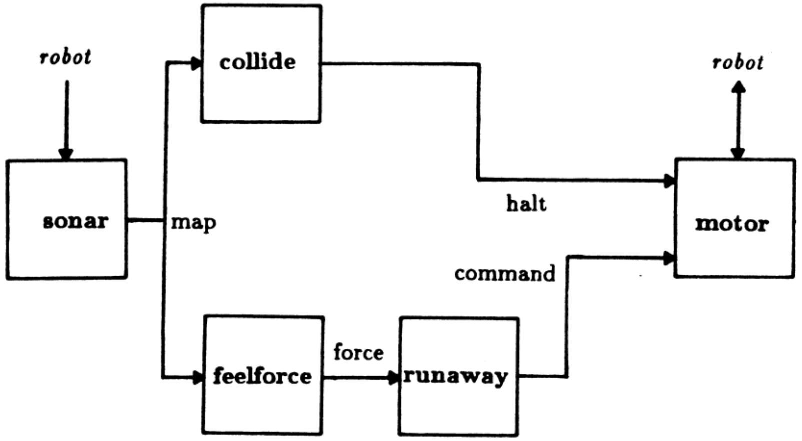
- Crossed IR beam identifies the object
- Extend arm till the finger beam is activated
- Hand can knock over the object
- Lift the hand

Case 3: Can is some distance away from hand

- Search for can near the surface than a few feet above
- Cruise the surface where can is placed
- Once IR detects a can, grab it

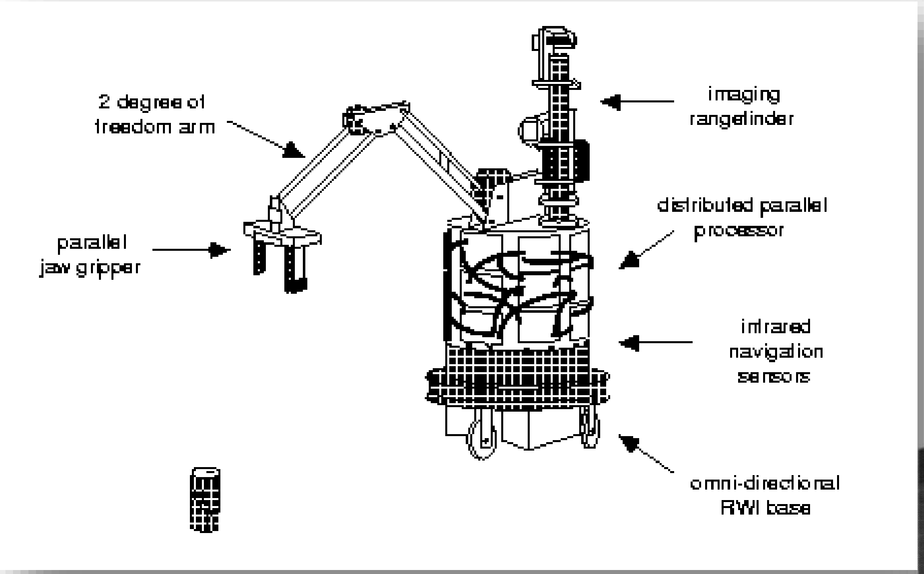


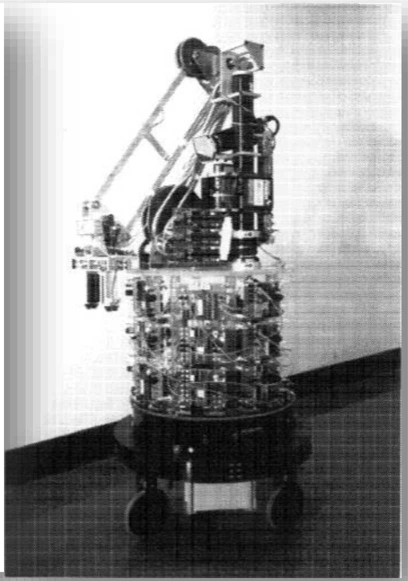
Example: Herbert



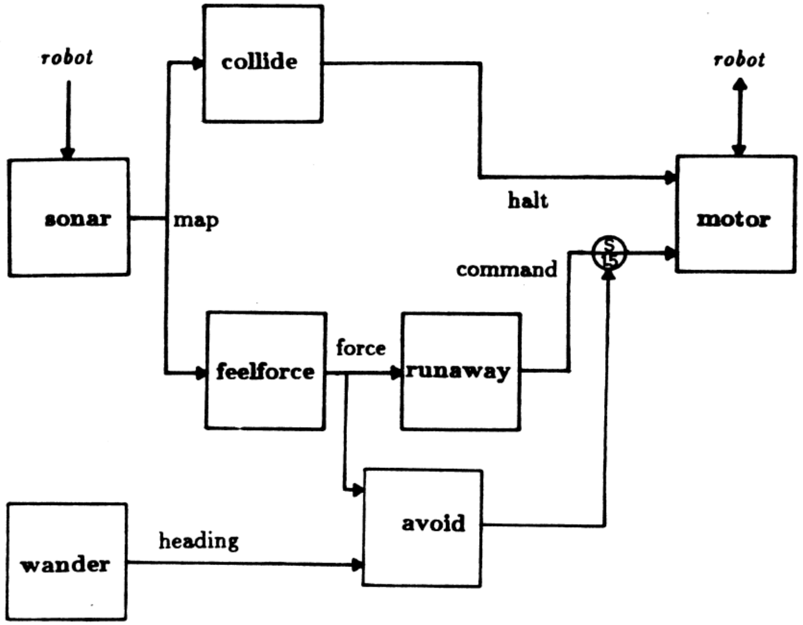
Herbert control architecture:

- Lev. 0: Runs away if approached, Avoids objects



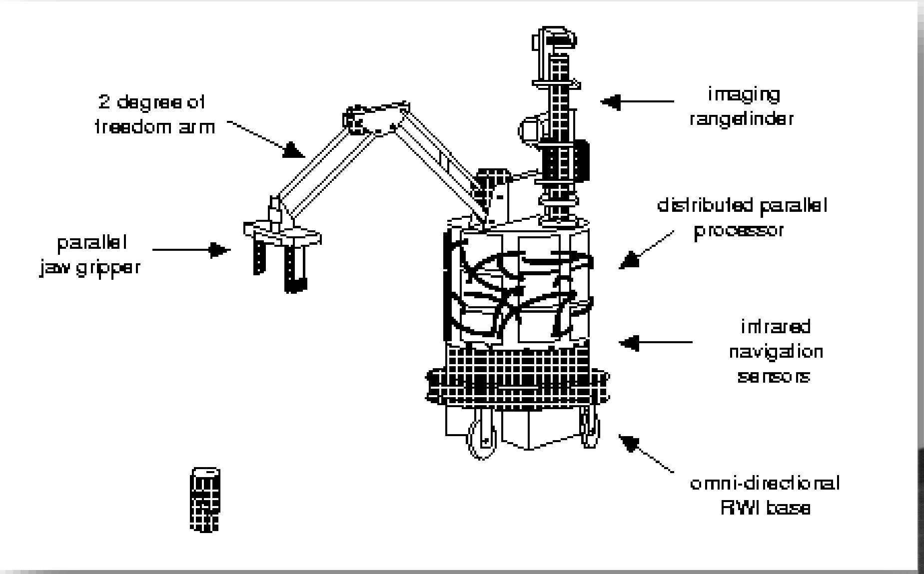


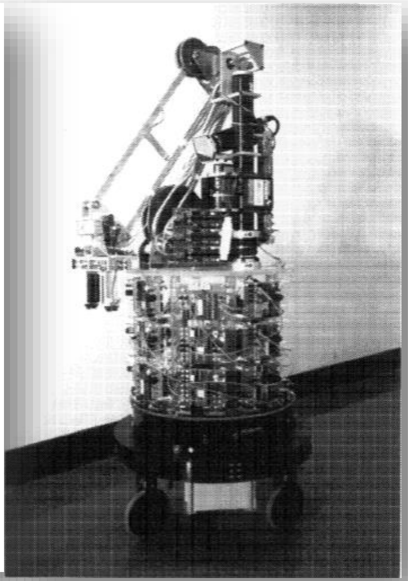
Example: Herbert



Herbert control architecture:

- Lev. 0: Runs away if approached,
  Avoids objects
- Lev. 1: Adds Wandering

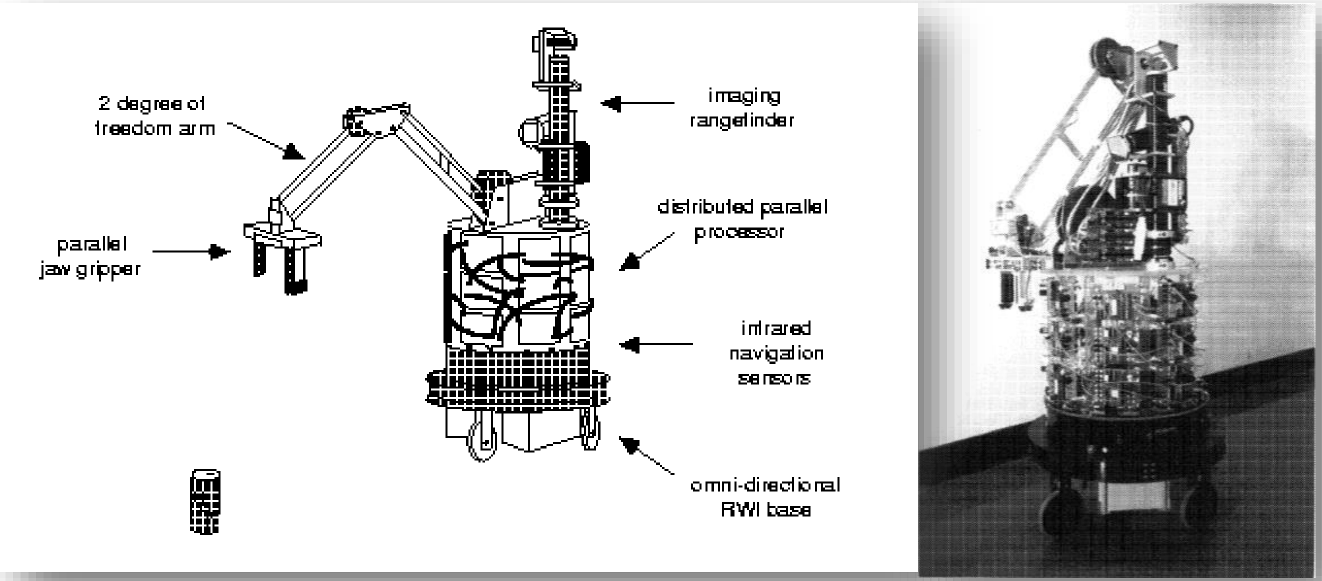


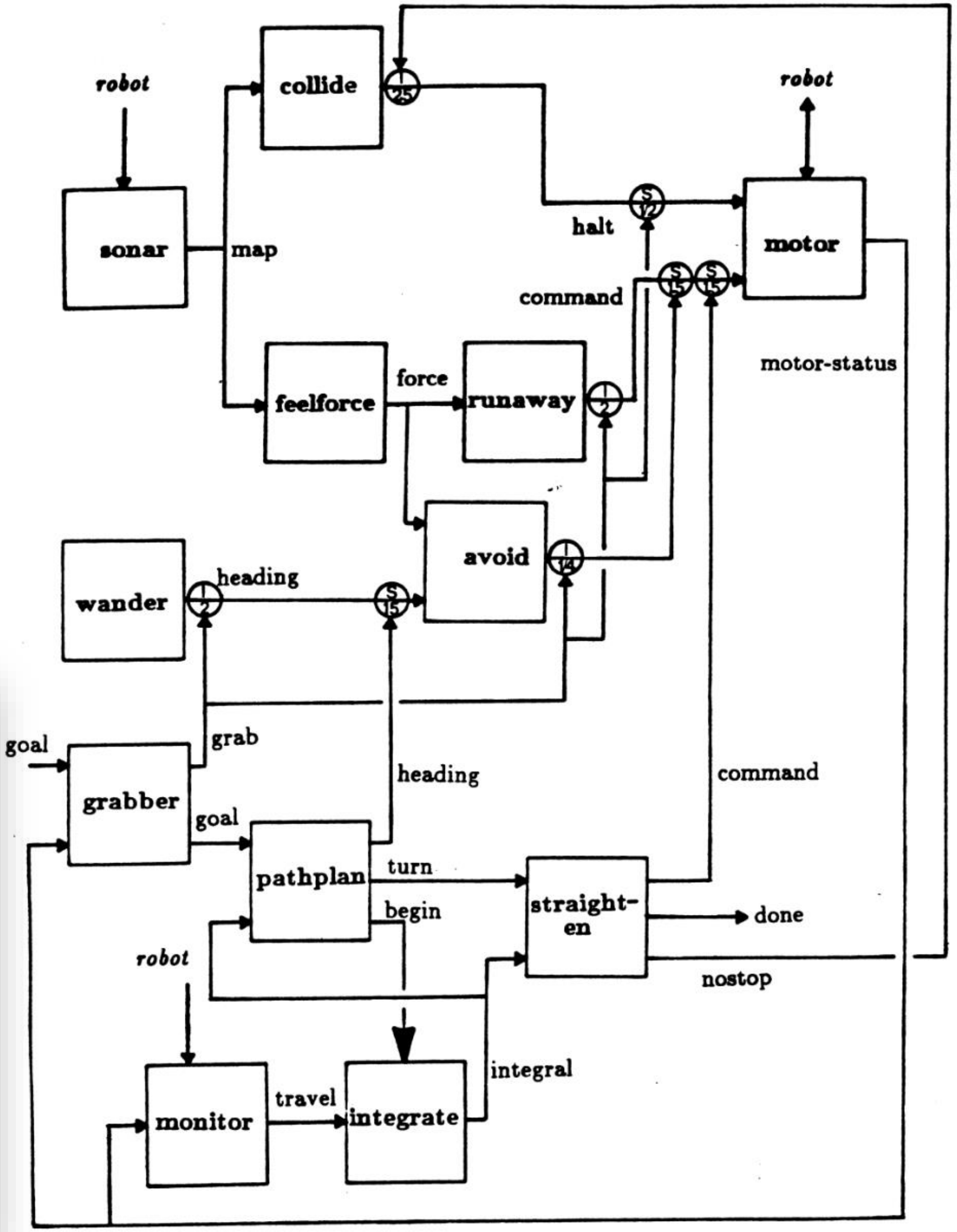


Example: Herbert

Herbert control architecture:

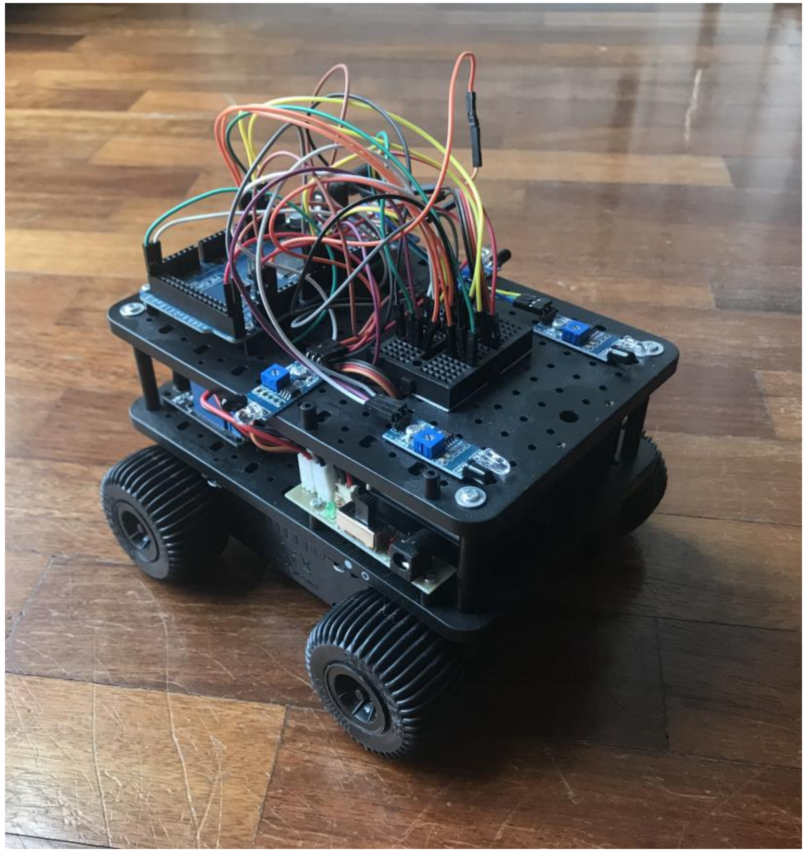
- Lev. 0: Runs away if approached,
  Avoids objects
- Lev. 1: Adds Wandering
- Lev. 2: Adds hallway following & grasping

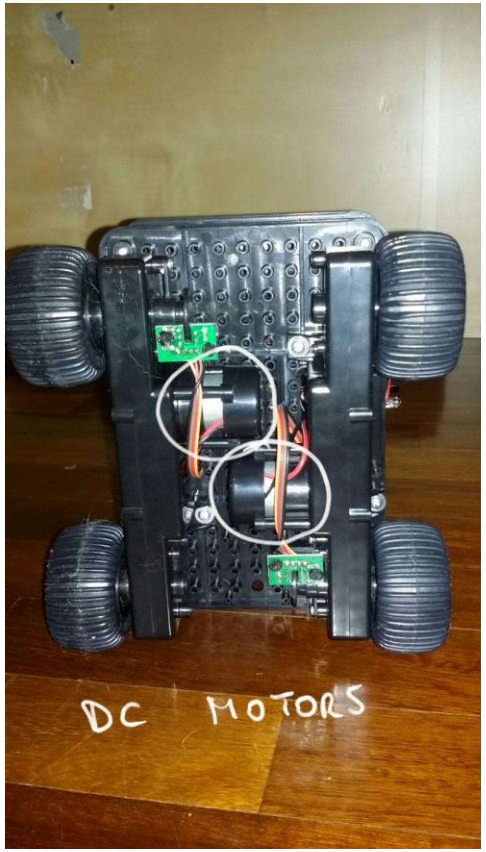


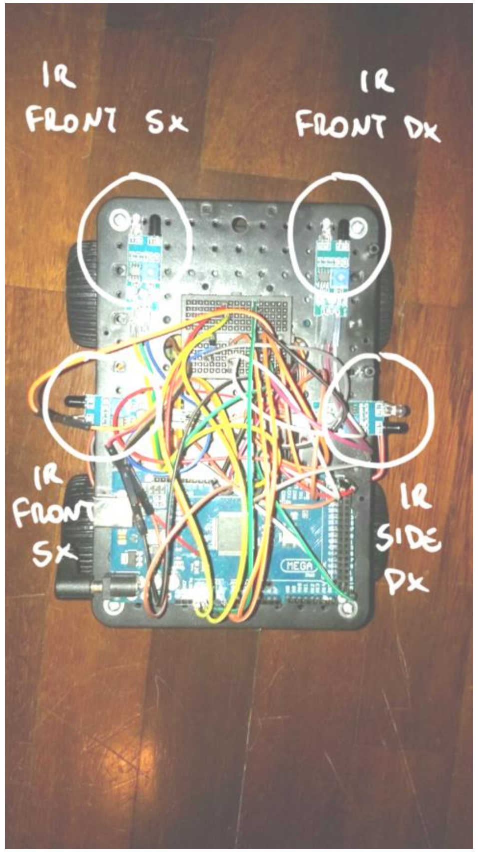


Roomba Like Behavior (Courtesy of Gabriel Manzoni)

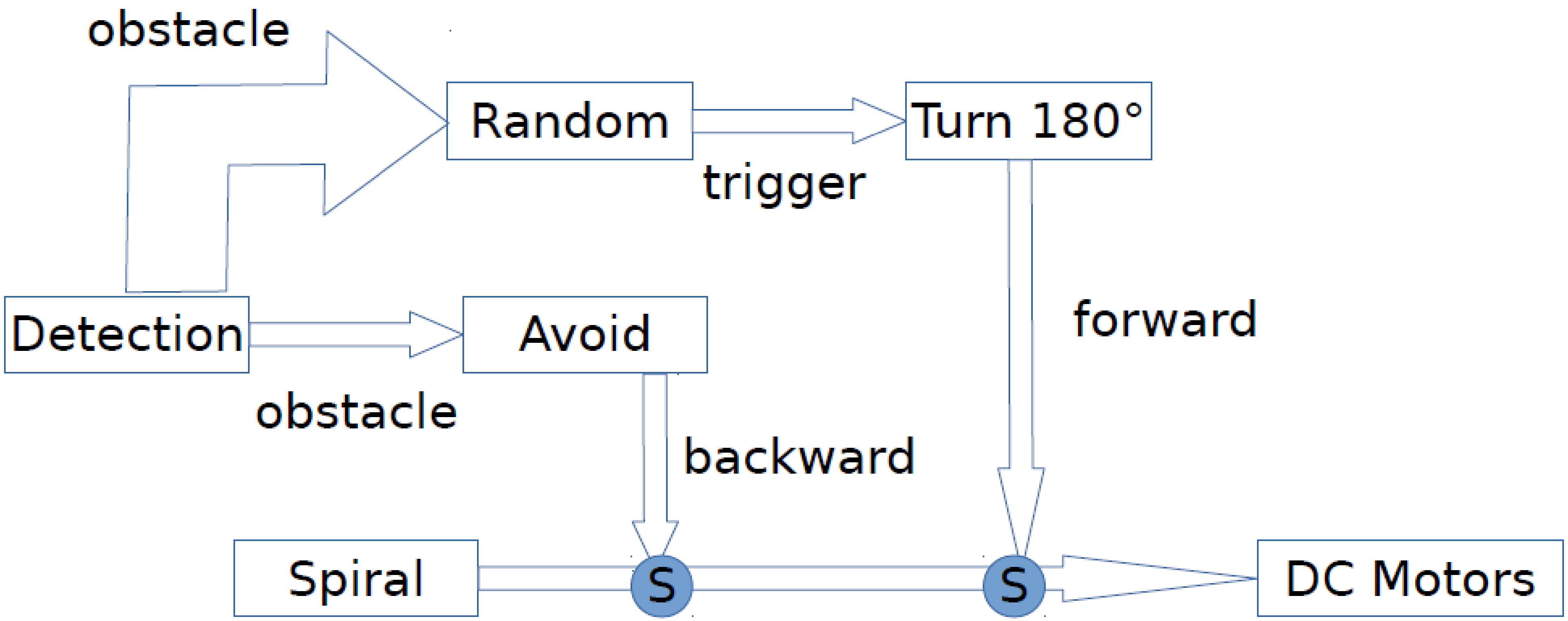
Project goal: Create a robot that emulates the behaviour of Roomba vacuum cleaner implementing various patterns



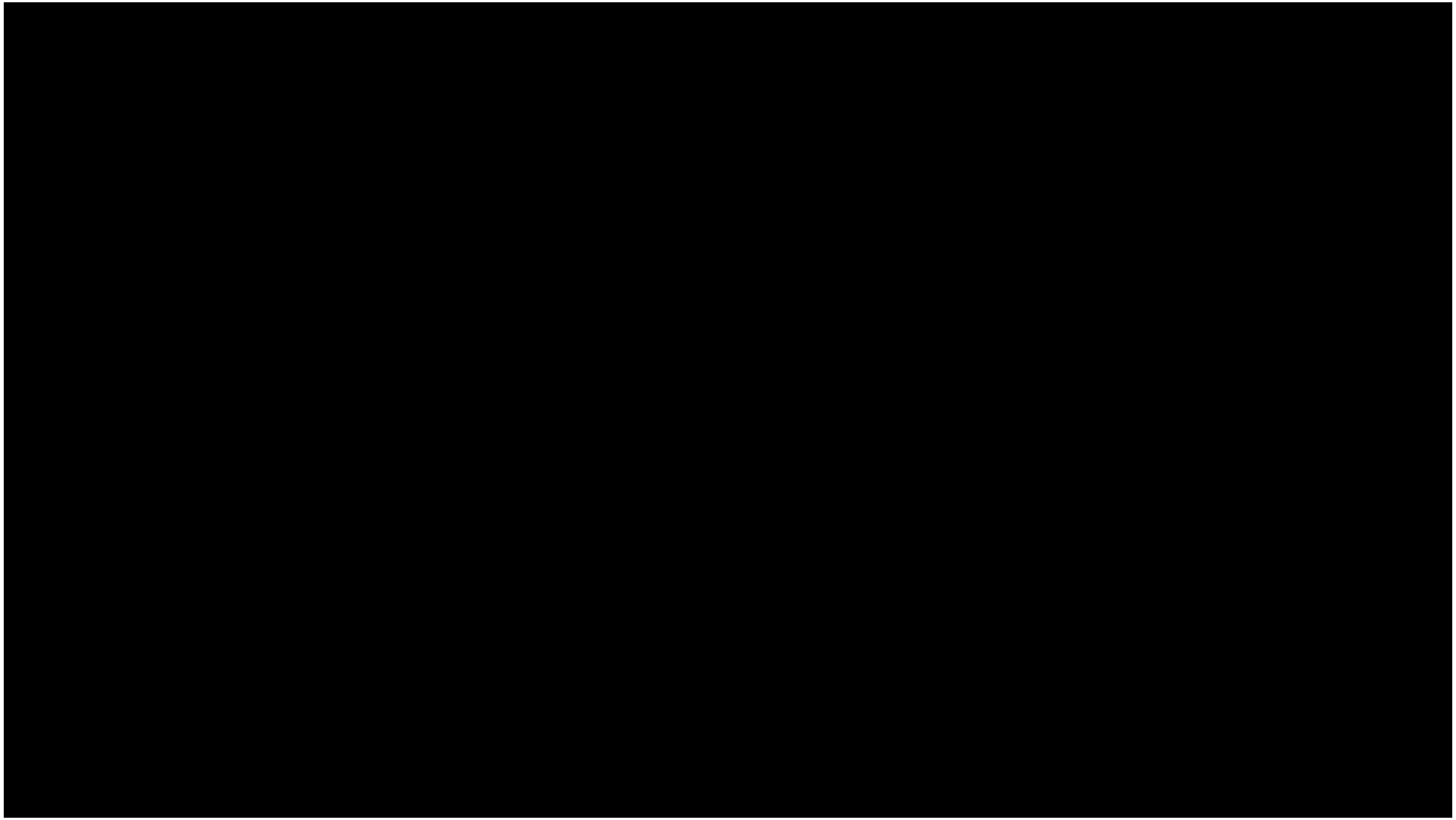




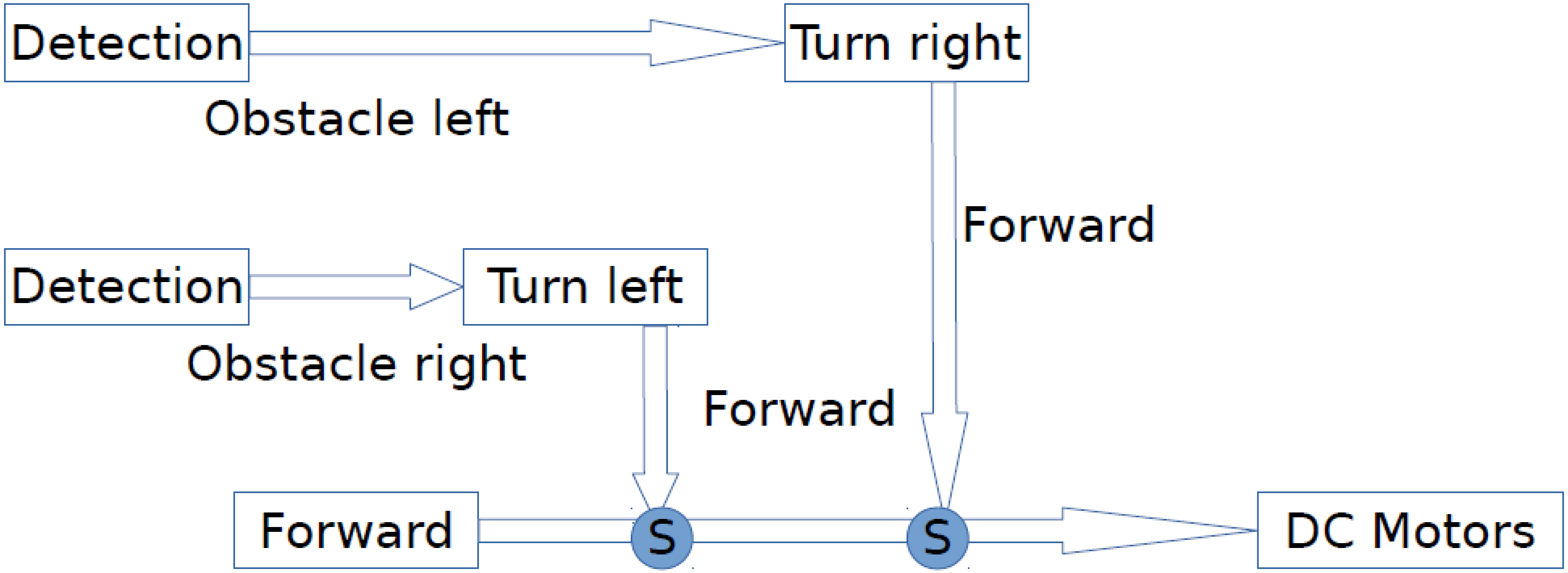
Behavior #1


obstacle
Random
trigger
Turn 180°
Detection
Avoid
forward
obstacle
backward
Spiral
S
S
DC Motors

Behavior #1



Behavior #2


Detection
Turn right
Obstacle left
Forward
Detection
Turn left
Obstacle right
Forward
Forward
S
S
DC Motors

Behavior #2

Random Switch between Behavior #1 and Behavior #2

Without obstacles											
Room 5 m x 5 m											
Movements	Random	Memory	Behaviour 1	Behaviour 2	Behaviour 3	Hours	Random	Memory	Behaviour 1	Behaviour 2	Behaviour 3
90.00%	20100	2987	15176	11670	13549	90.00%	5.6	0.8	4.2	3.2	3.8
95.00%	25550	3360	26879	18433	19934	95.00%	7.1	0.9	7.5	5.1	5.5
99.00%	38854	3599	48504	33789	31645	99.00%	10.8	1.0	13.5	9.4	8.8
100.00%	69142	4479		51612	39123	100.00%	19.2	1.2	0.0	14.3	10.9

With Obstacles											
10 Obstacles	Random size										
Movements	Random	Memory	Behaviour 1	Behaviour 2	Behaviour 3	Hours	Random	Memory	Behaviour 1	Behaviour 2	Behaviour 3
90.00%	20124	10064	19729	15171	17614	90.00%	5.6	2.8	5.5	4.2	4.9
95.00%	26989	12036	34943	23963	25914	95.00%	7.5	3.3	9.7	6.7	7.2
99.00%	41521	17218	63055	43926	41139	99.00%	11.5	4.8	17.5	12.2	11.4
100.00%	79887	21543		67096	50860	100.00%	22.2	6.0	0.0	18.6	14.1

100 Obstacle	Random size										
Movements	Random	Memory	Behaviour 1	Behaviour 2	Behaviour 3	Hours	Random	Memory	Behaviour 1	Behaviour 2	Behaviour 3
90.00%	28319	11137	21702	16688	19375	90.00%	7.9	3.1	6.0	4.6	5.4
95.00%	39241	13725	38437	26359	28506	95.00%	10.9	3.8	10.7	7.3	7.9
99.00%	70041	17223	69361	48318	45252	99.00%	19.5	4.8	19.3	13.4	12.6
100.00%	141516	21576		73805	55946	100.00%	39.3	6.0	0.0	20.5	15.5

Subsumption pros and cons

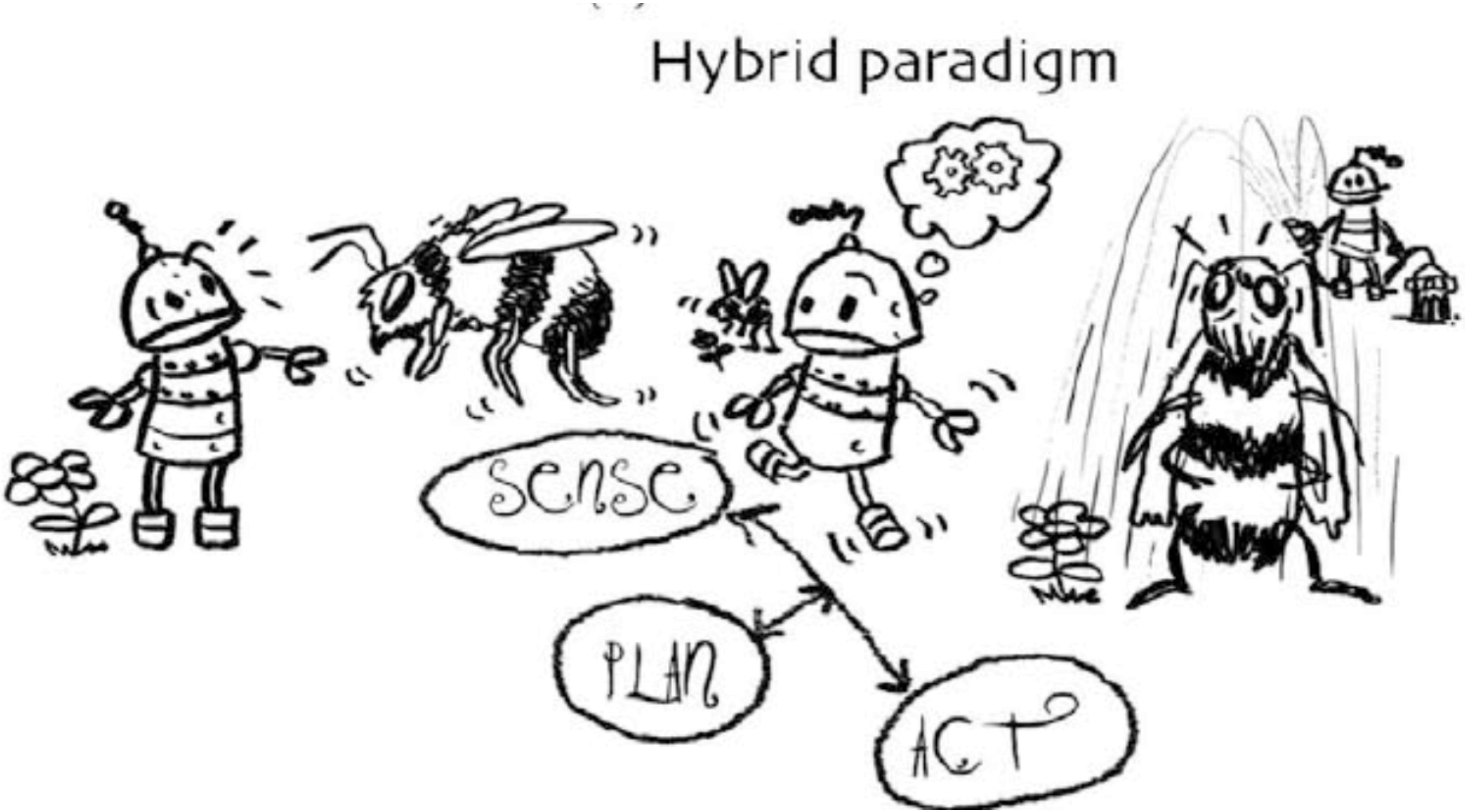
Pros …

- Provides a way to incrementally build/test a complex mobile robot control system
- Supports parallel computation in a straightforward, intuitive way
- Avoids centralized control; relies on self-centered and autonomous modules
- Leads to more emergent behavior: “Complex (and useful) behavior may simply be the reflection of a complex environment”

Cons ...

- In the development of an individual, new representational structures are developed in response to the environment, not added by an experimenter.
- It would be more impressive if the robot learnt new behavior modules in response to the environment.
- Emphasis in this approach on reacting to the environment, but representations are needed for more complex tasks.

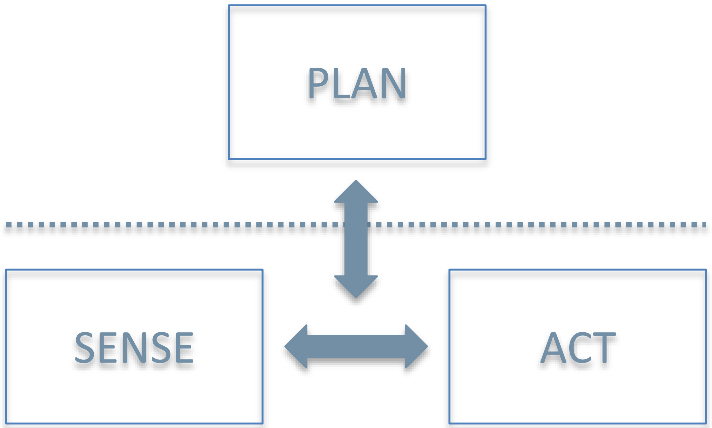
The Best of Two Worlds



Organization: Plan, Sense-Act

Reactive Paradigm was the major trend by the end of the 1980’s, however it does not:

- Remember the state of the robot/world
- Plan optimal trajectories
- Make maps
- Monitor its own performance
- Select the best behaviors for a task



To differentiate from planning, term *deliberative* was coined.

Should planning be reintroduced?

- Reactive functions for low level control (PRESENT with a bit of duration)
- Deliberation for higher level tasks (reason about PAST project into the FUTURE)

Plan, Sense-Act Sensing Organization

The Map (World Model) performs integration

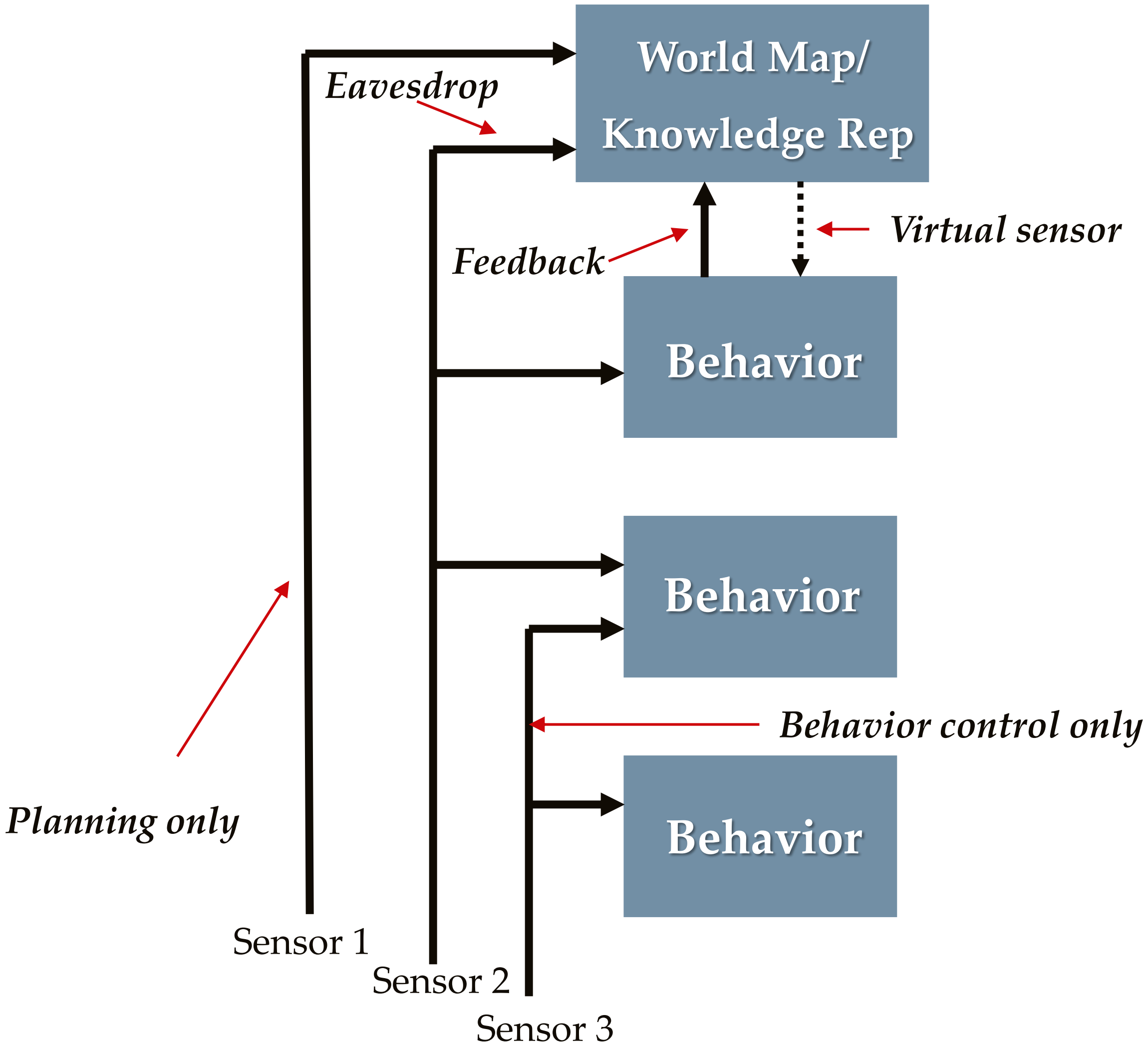
- Can have its own sensors
- Can “eavesdrop” on other sensors
- Can act as “virtual” sensor

Reactivity components

- No global knowledge
- Work with sensors and actuators

Deliberation components

- Global knowledge
- Work with symbols



48

Common Components

Sequencer: generates a sequence of behaviors

Resource Manager: allocates resources to behaviors

Cartographer: creates, stores, maintains, accesses map information

Mission Planner: interact with human and create a plan to achieve a goal

Performance Monitor/problem solver: determines whether the robot is making progress toward its goal

Three forms of Architectural Styles

- Managerial (division of responsibility as in business): AuRA, SFX
- State Hierarchies (strictly by time scope): 3T
- Model-Oriented (Model serve as virtual sensors): Saphira, TCA

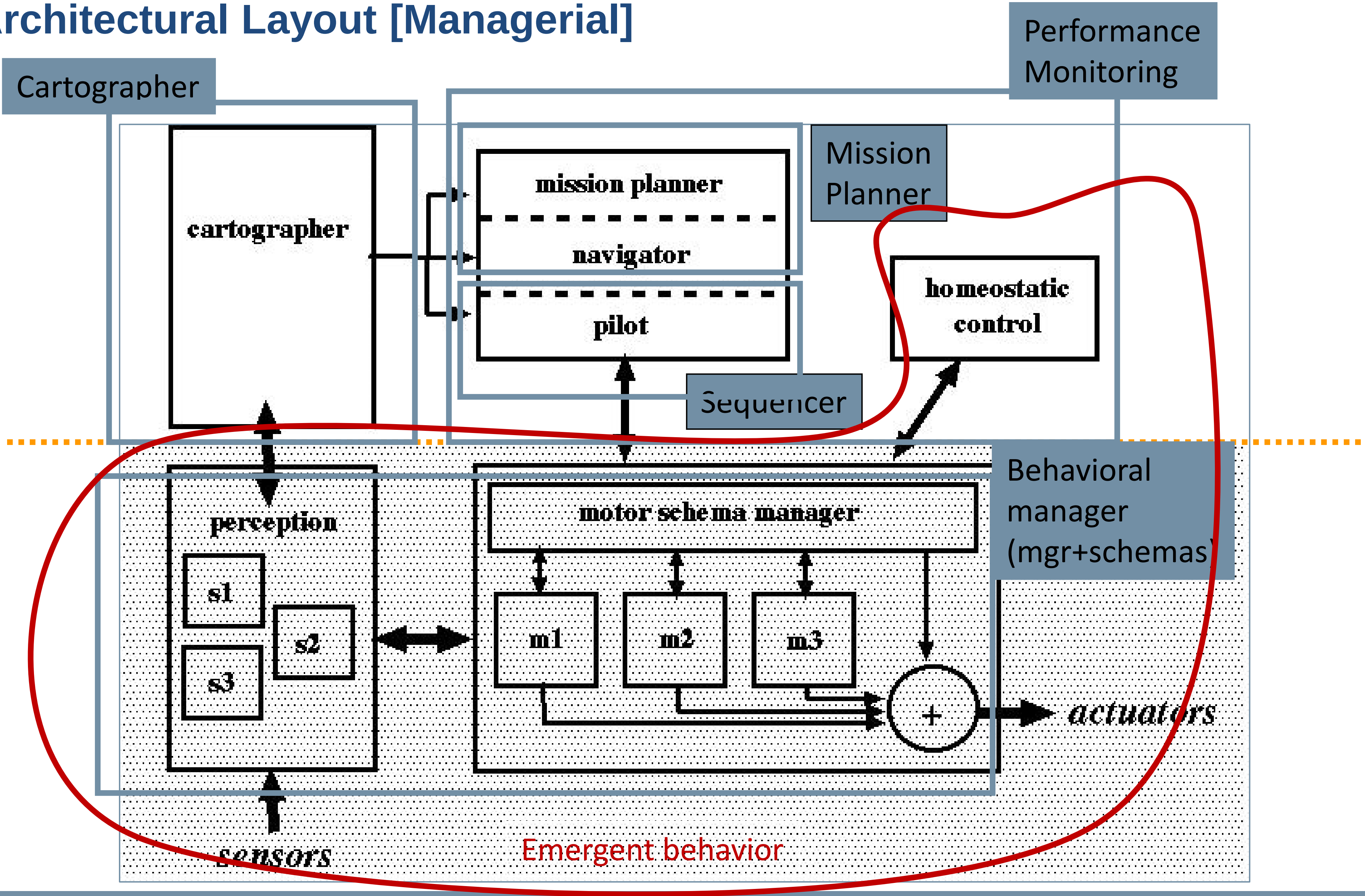
AuRA (Autonomous Robot Architecture) [Managerial]

Consists in 5 subsystems

- sensor: acquire and interpret data
- motor: motor schema behavior control
- cartographer: all map making, reading functions
- planner: responsible for mission and task planning
- homeostatic control: modify the relationship between behaviors by changing the gain as a function of robot or other constraints

Ron Arkin, Georgia Institute of Technology

AuRA Architectural Layout [Managerial]



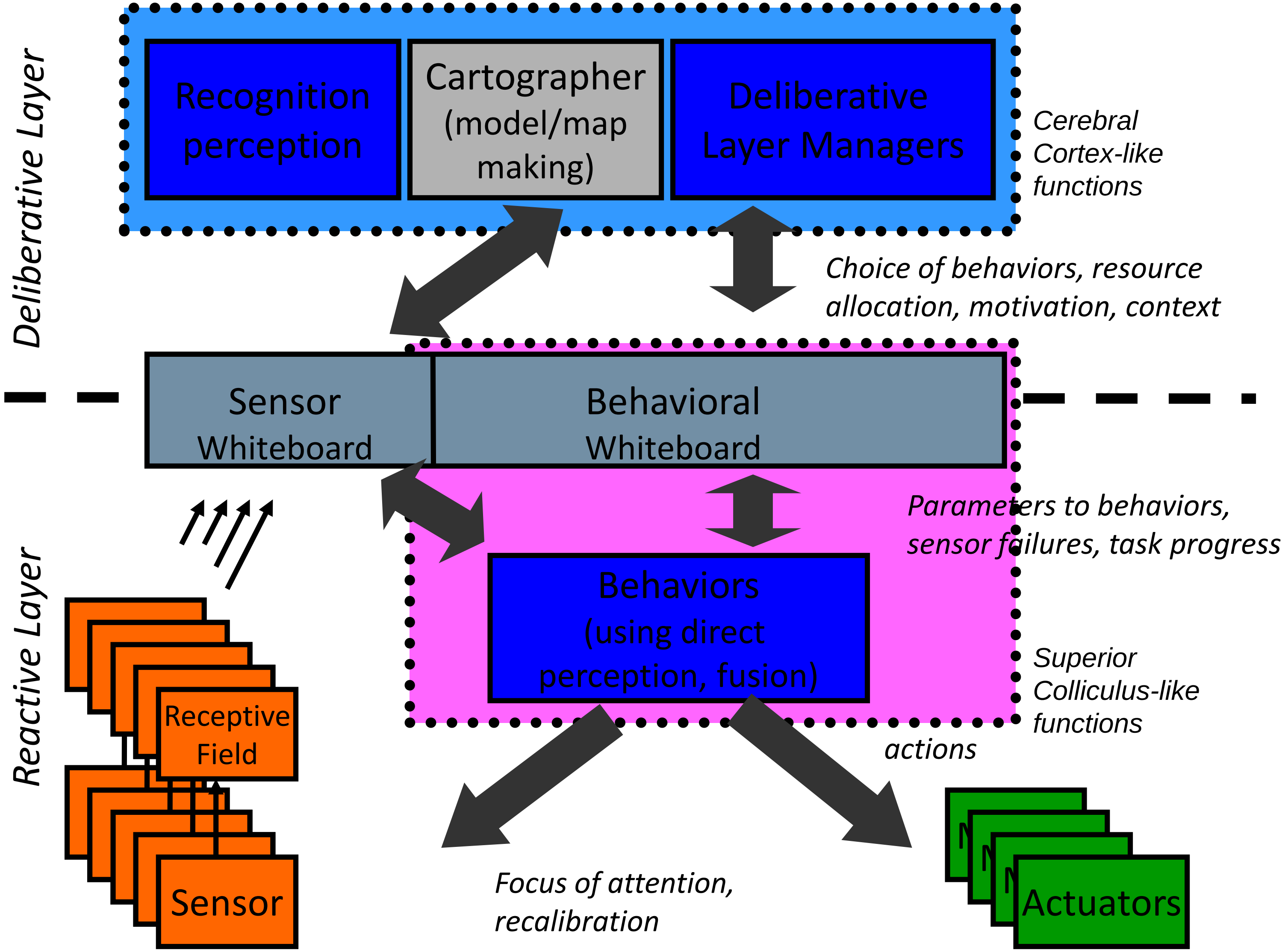
SFX (Sensor Fusion Effects) [Managerial]

Extension to AuRA adding modules to specify how sensing and handling sensor failure (Murphy 2000).

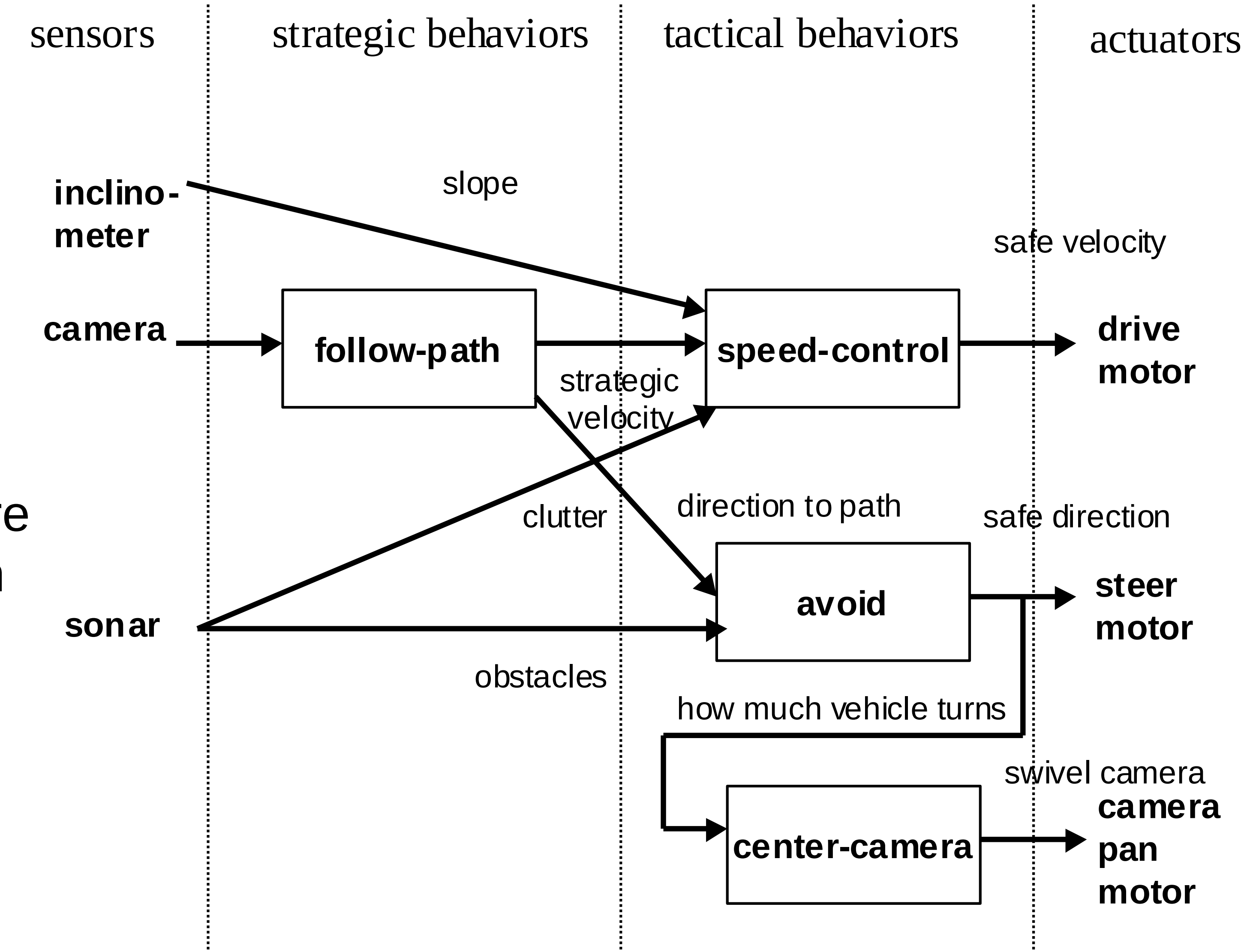
- Focus on sensing
- Biomimetic organization
- Deliberative layer consists of managerial agents
- Reactive layer has *tactical behaviors*

Deliberative layes determine the best allocation of effect, sensing resource and perceptual schema.

SFX (Sensor Fusion Effects)



Tactical Behaviors



Tactical behavior serves as filter on strategic commands to ensure to robot acts in a safe manner in as close accordance with the strategic intent as possible

Summary of Managerial Architectures

How does the architecture distinguish between reaction and deliberation?

- Deliberation: global knowledge or world models, projection forward or backward in time
- Reaction: behaviors which have some past/persistence of perception and external state

How does it organize responsibilities in the deliberative portion?

- Hierarchy of managerial responsibility, managers may be peer software agents

How does overall behavior emerge?

- From interactions of a set of behaviors dynamically instantiated and modified by the deliberative layer
- Assemblages of behaviors

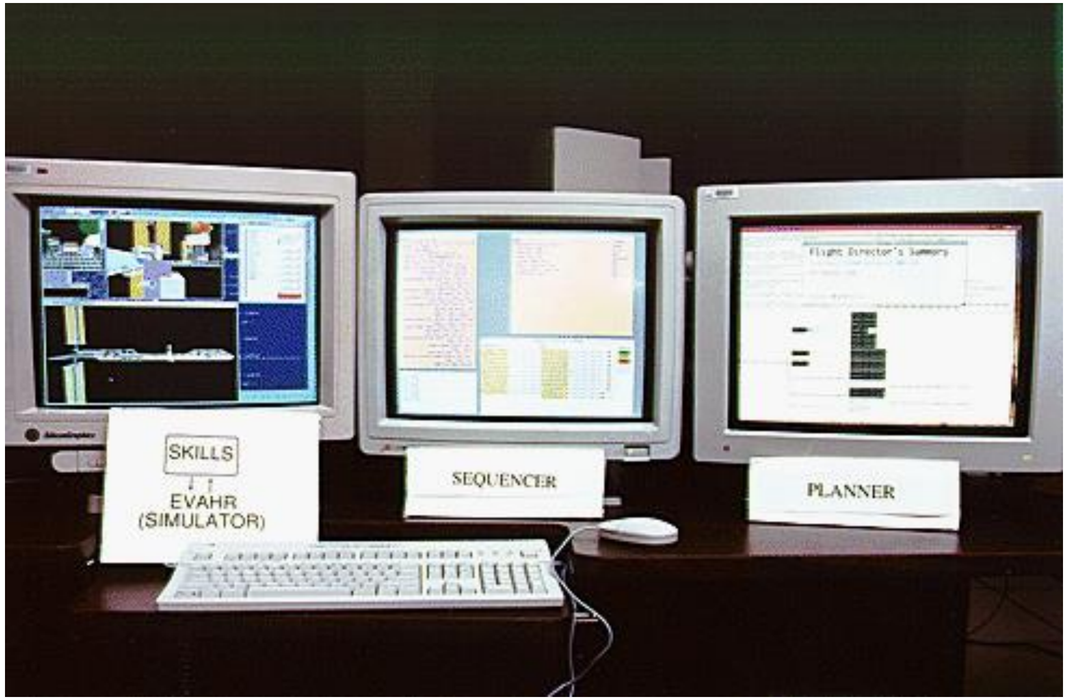
3T Architecture [State Hierarchy]

3 Tiers Architecture proposed and used extensively at NASA

- Deliberative: setting goal and strategic plans
- Reactive: in this layer the skills have associated events to verify explicitly an action has had the correct effect
- Sequencer: select a set of primitive behaviors develop a task network (reactive planning)

Built by merging

- Subsumption variation (Gat, Bonasso),
- RAPs (Firby)
- Vision (Kortenkamp)



TRAC Labs (NASA JSC)

3T Architecture [State Hierarchy]

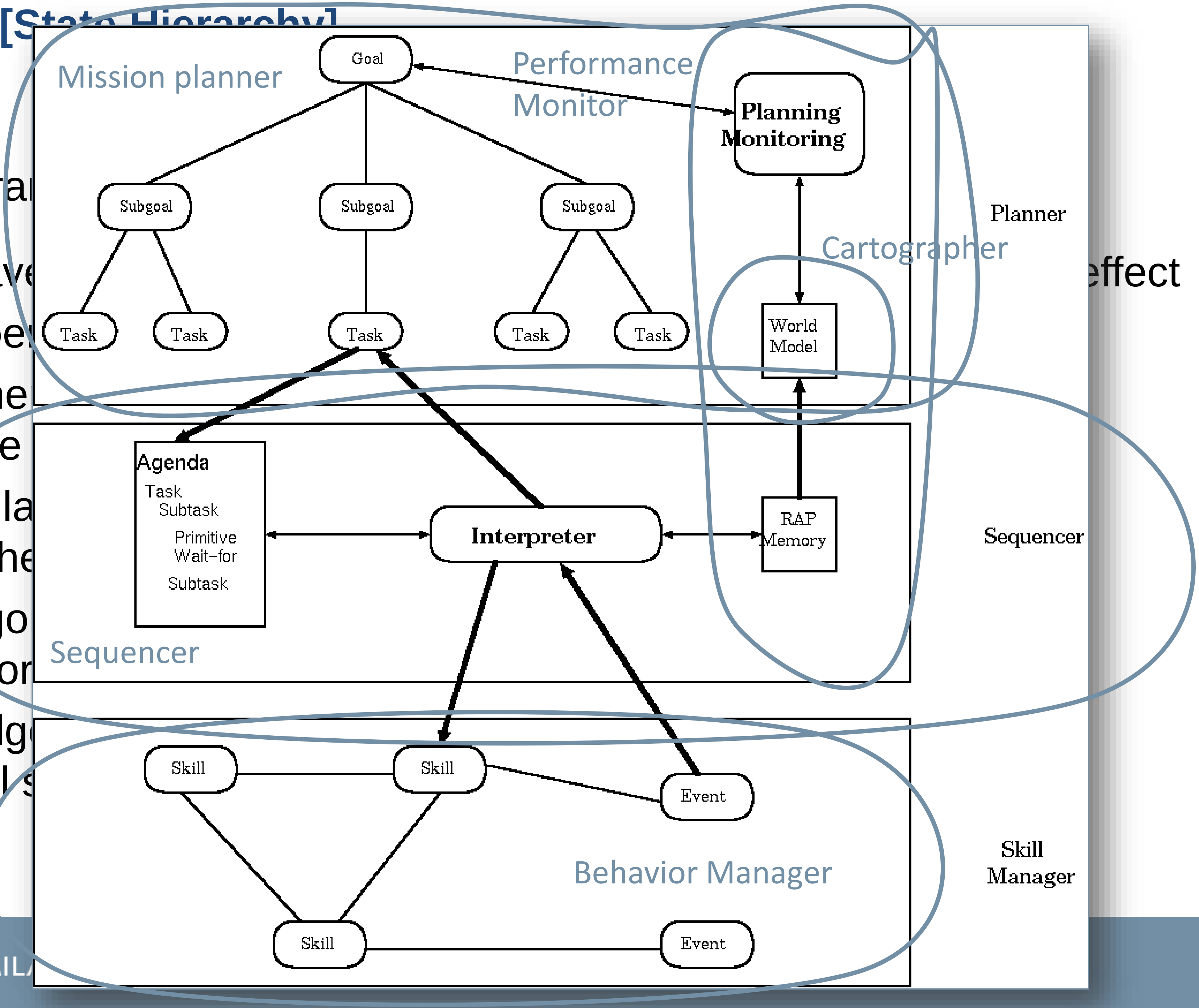
Execution is arranged by time and execution rate

- Skills have associated events, to verify that an action had correct effect
- Skills operate only in the Present
- Components of the sequencer layer operate on state information about the Past, as well as Present
- Planner layer works state information about the Past and Present to plan the Future
- Slow algorithms are in the Planner,
Fast algorithms go into the Skill Manager Layer
- Vision algorithms were placed in the Planner despite their low-level sensing functions

3T Architecture [State Hierarchy]

Execution is arra

- Skills have ... effect
- Skills ope
- Compone about the
- Planner la to plan the
- Slow algo Fast algor
- Vision alg low-level s



Summary on State Hierarchy Architectures

How does the architecture distinguish between reaction and deliberation?

- Deliberation: requires PAST or FUTURE knowledge
- Reaction: behaviors purely reflexive with local, behavior specific, PRESENT knowledge

How does it organize responsibilities in the deliberative portion?

- By internal temporal state
    - PRESENT (controller)
    - PAST (sequencer)
    - FUTURE (planner)
- By speed of execution

How does overall behavior emerge?

- From generation and monitoring of a sequence of behaviors
- Assemblages of behaviors called skills
- Subsumption

Model-Oriented Architectures

Top-down, symbolic flavor:

- Symbolic manipulation around a global world model
- World model supply perception with virtual sensors

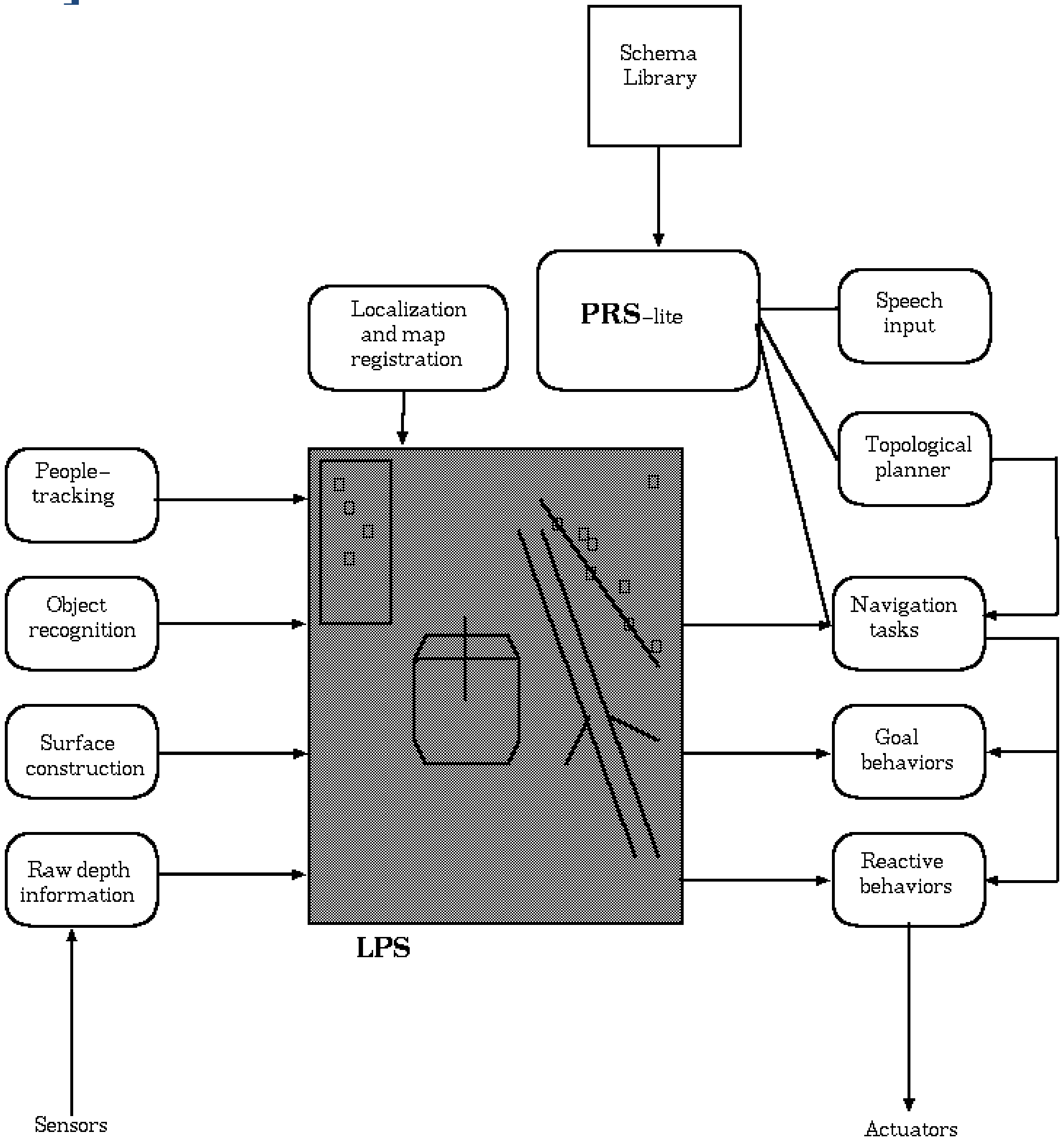
Different than the hierarchical model:

- model restricted to labeling objects of interest like hallway, door, etc
- perceptual processing is distributed and asynchronous
- sensor errors and uncertainty can be filtered using sensor fusion over time to improve performance
- increase in processor speed and optimizing compilers solved the processing bottleneck

Saphira Architecture [Model Oriented Architecture]

Developed at SRI by Konolige, Myers, Saffioti

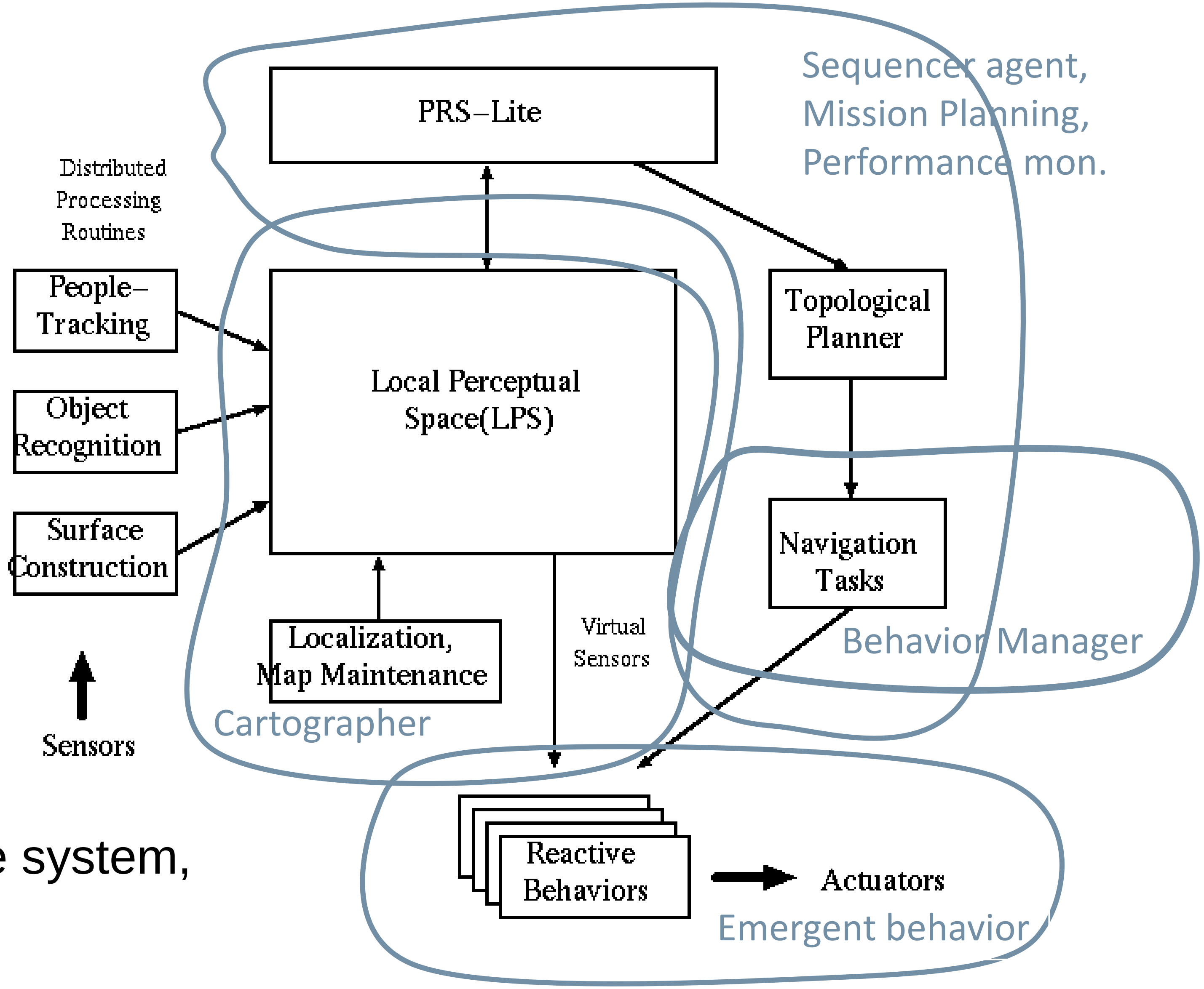
- Comes with Pioneer robots
- Behaviors produce fuzzy outputs fuzzy logic combines them
- Has a global rep called a ***Local Perceptual Structure*** to filter noise
- Instead of RAPs, uses PRS-Lite



Saphira and PRS - Procedural Reasoning System

Reactivity in planning:

- Postponement of the elaboration of plans until it is necessary
- plans are determined continuously in reaction to the current situation
- plans in progress can be interrupted and abandoned at any time
- plans represent the robot's desired behavior
- a symbolic plan always drives the system,

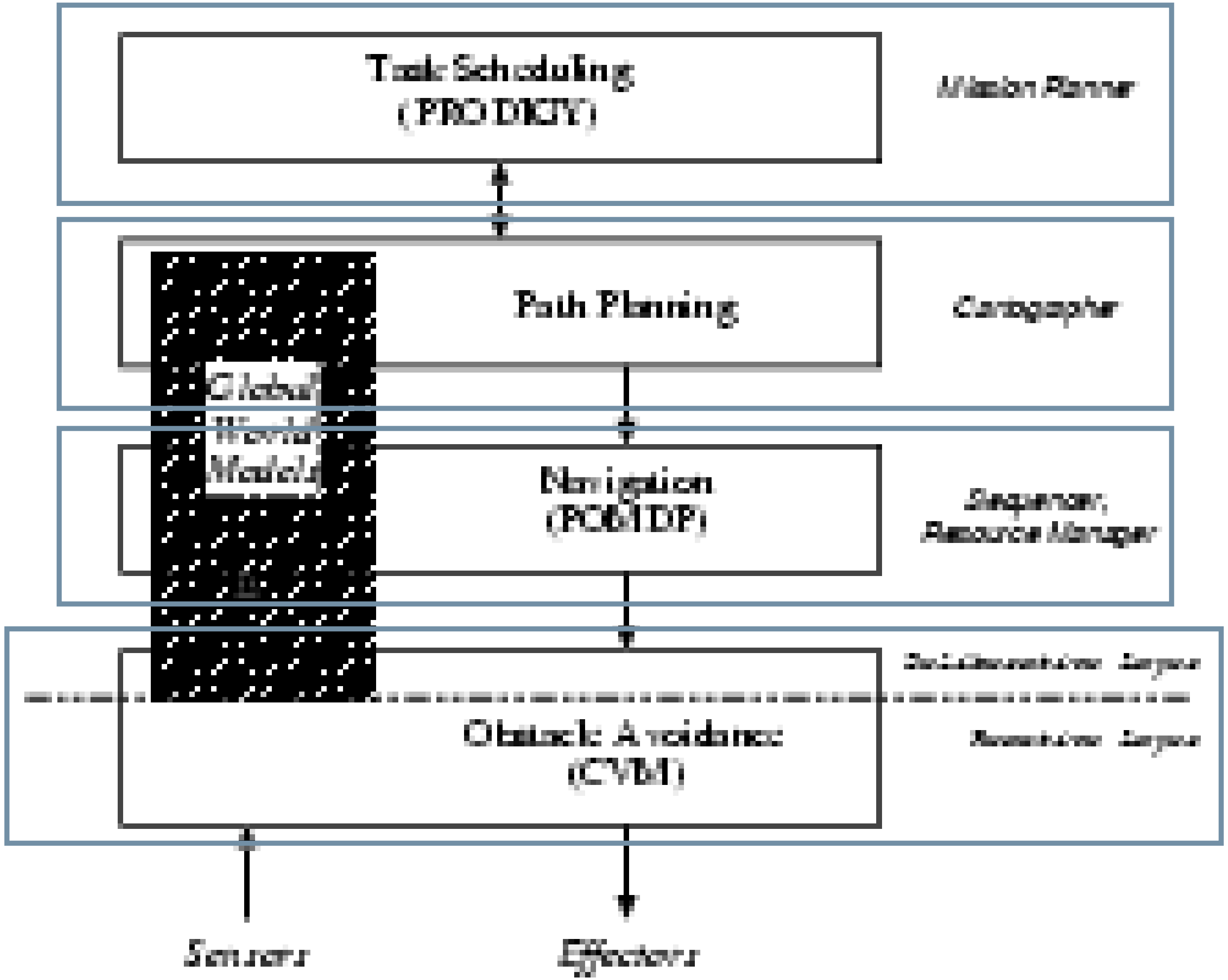


Task Control Architecture [Model Oriented Architecture]

Developed by Reid Simmons, used extensively by CMU Field Robotics Projects and NASA's Nomad, Ambler, Dante

- Closer to an operating system architecture with Tasks instead of behaviors
- Uses dedicated sensing structures called evidence grids as distributed global world model
- Task Scheduling Layer (using Prodigy planner) determines the task flow, interacts with the user, determines the goals and order of execution
- Navigation by a Partially Observable Markov Decision Process (POMDP)
- Obstacle Avoidance Layer uses a curvature-velocity to respond with a smooth trajectory

Task Control Architecture [Model Oriented Architecture]



Model-Oriented Architectures

How does the architecture distinguish between reaction and deliberation?

- Deliberation: anything relating a behavior to a goal or objective
- Reaction: behaviors are “small control units” operating in present, but may use global knowledge as if it were a sensor (*virtual sensor)*

How does it organize responsibilities in the deliberative portion?

- Model of the world and state of the robot
- Hierarchical Paradigm with global world model but virtual sensors
- Deliberative functions

How does overall behavior emerge?

- From generation and monitoring of a sequence of behaviors
- Voting or fuzzy logic for combination