UNIVERSITÀ DEGLI STUDI DI PADOVA

ARCHITECTURES FOR AUTONOMOUS ROBOTS

Prof. Emanuele Menegatti

Intelligent Robotics Course

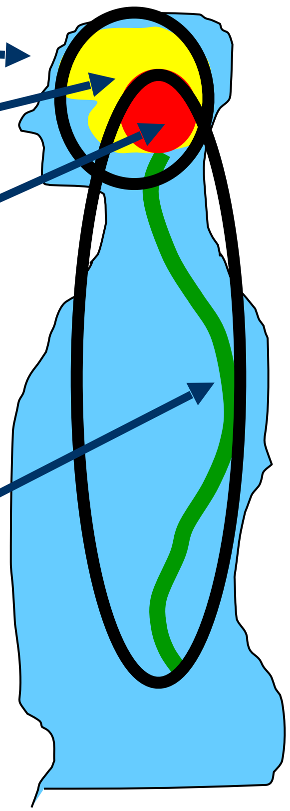
The Operational Architecture of biologic intelligence

Interaction:
Persona and teaming

Deliberative loop:
Reasoning over symbols

Converting sensor data into symbols
(Key Functionality)

Reactive loop:
Skills and responses



Interaction Layer

Deliberative Layer

Reactive (or Behavioral) Layer

Can we make this more tangible?

Classical Primitives for Robot Intelligence

SENSE

PLAN

ACT

LEARN

Problems:

- **How to combine these blocks?**
- **What to insert in these blocks?**

Answer:

- **Use robotic paradigms**

Robotics's Paradigms

- **A paradigm is a** philosophy or a set of hypotheses and / or techniques that characterise an approach to a class of problems.
- Three paradigms for organising intelligence:
  - Hierarchical
  - Reactive
  - Deliberative / reactive hybrid
  - (NEW) Deep Learning Robotics (???)
- Two ways to describe them, through:
  - the relationships between the three primitives of robotics: SENSE, PLAN, ACT
  - how sensory data is managed and distributed in the system

The classical AI Approach:

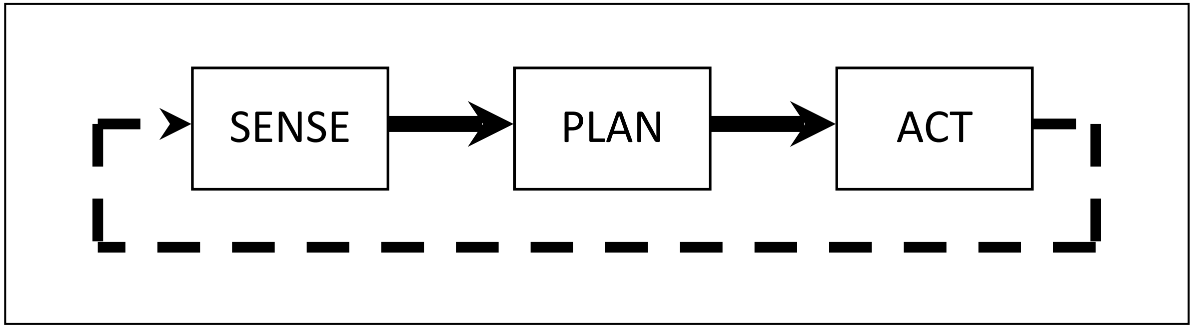
John McCarty, and Nils Nilsson, Stanford, USA

Bernard Meltzer, Edinburg, UK

HYERARCHICAL PARADIGM

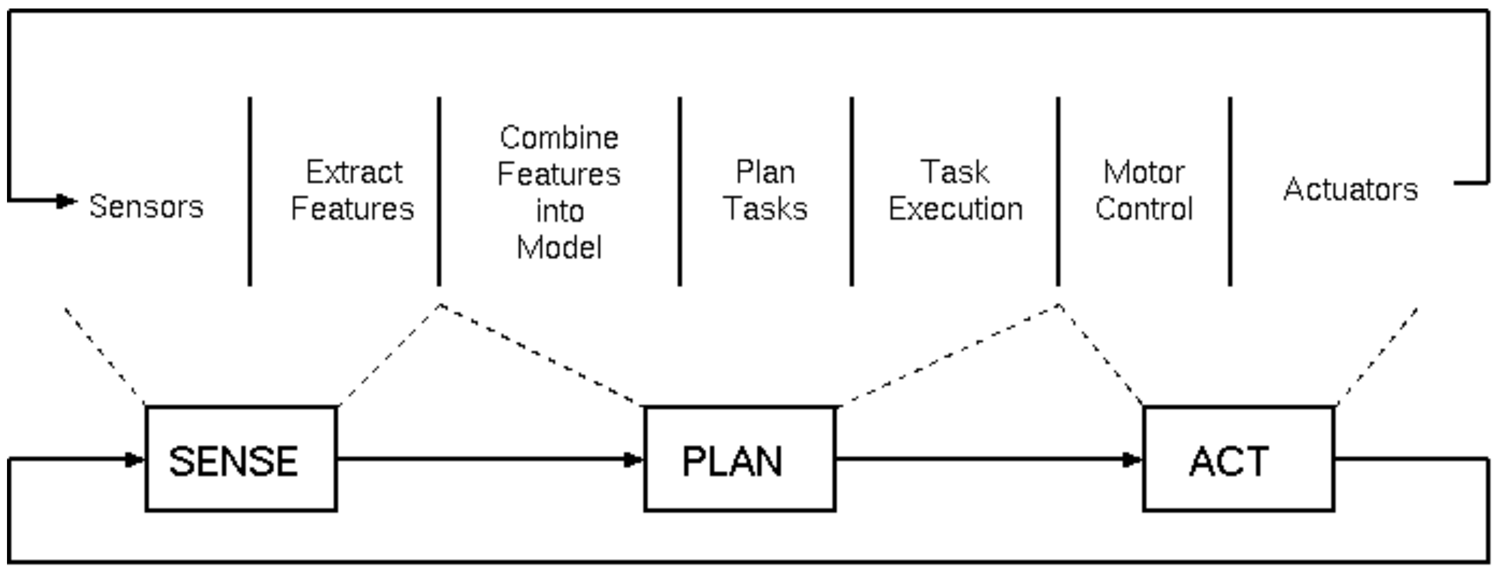
- Classical Planning
  - Predicate Logic
  - Logical Actions
  - Planning Algorithms
- Task Policies
- Integrate Task and Motion Planning

Hierarchical Paradigm: Sequence of 3 Primitives



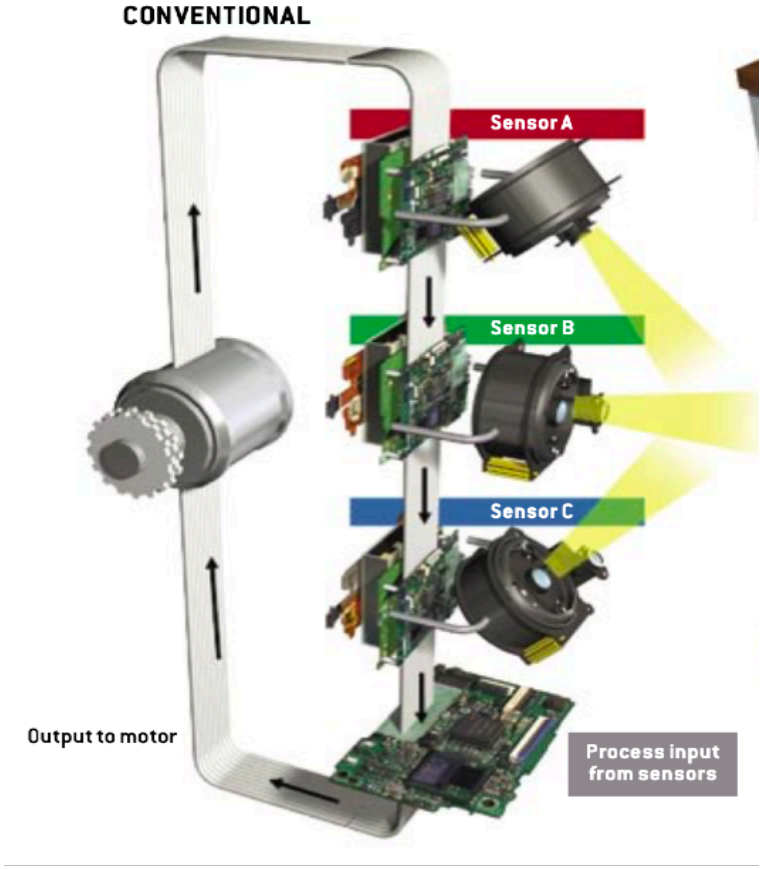
Adapted from R.Murphy

Hierarchical Paradigm



World model is a fused global data structure.
It combines:
1. A priori representation
2. Sensed info
3. Cognitive understanding

Hierarchical Paradigm

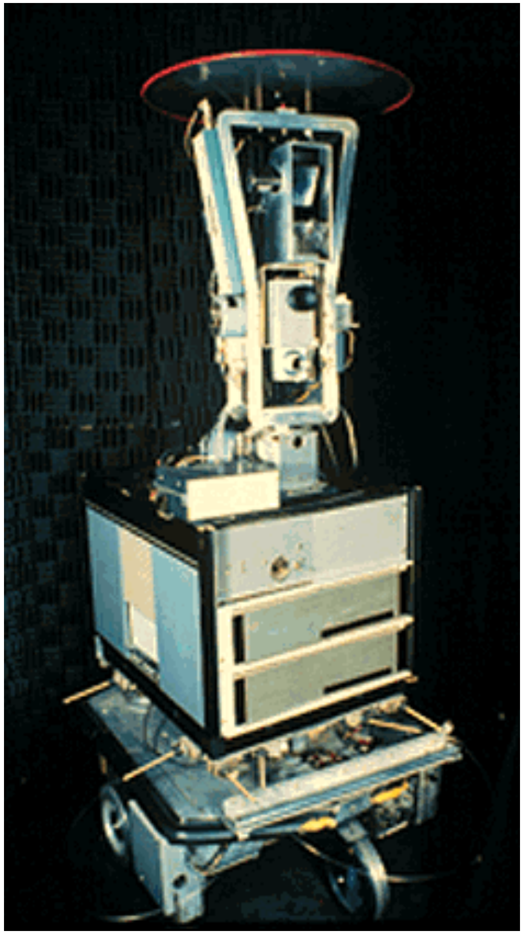


Hyerarchical paradigm

Primitive	Input	Output
SENSE	Sensor data	Sensed Information
PLAN	Information (sensed and/or cognitive)	Directives
ACT	Sensed information or directives	Actuator commands

Hyerarchical paradigm

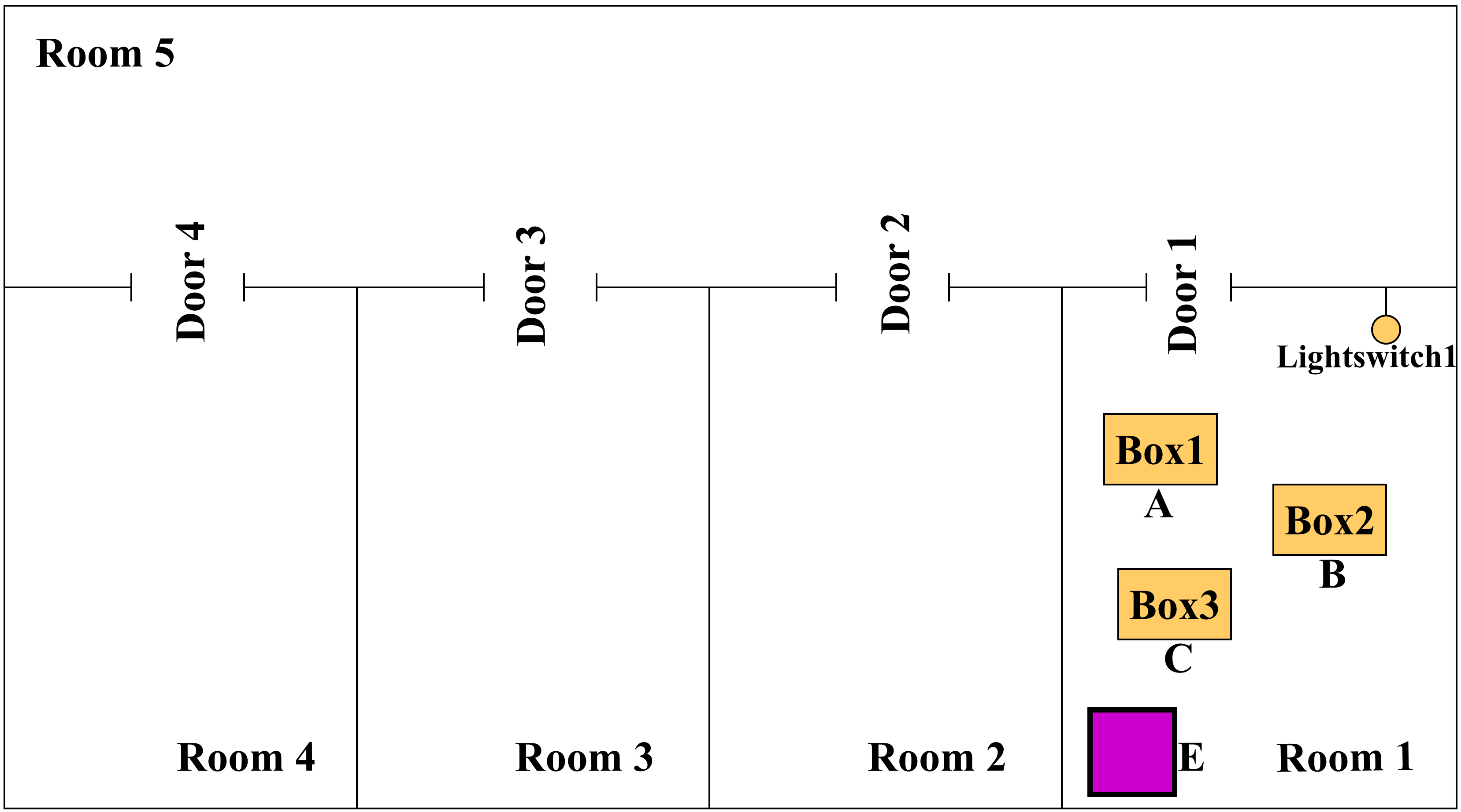
- First AI robot
- Built by SRI (Stanford Research Institute) for DARPA 1967-9
- Used ***Strips*** as main algorithm for controlling what to do
- Dominates robotics until 1980



Hierarchical Paradigm: Notable Systems Architectures

- **STRIPS/GPS** (*Nilsson*)
  - Not used anymore, but did spawn planning industry
  - Shakey

STRIPS e Shakey-1



STRIPS E SHAKEY-2

- Initial World Model:
  - CONNECTS(Door1,Room1,Room5)
  - CONNECTS(Door2,Room2,Room5)
  - CONNECTS(Door3,Room3,Room5)
  - CONNECTS(Door4,Room4,Room5)
  - AT(Box1,A)
  - AT(Box2,B)
  - AT(Box3,C)
  - AT(Lightswitch1,D)
  - ATROBOT(E)
  - TYPE(Box1,Box)
  - TYPE(Box2,Box)
  - TYPE(Box3,Box)
  - TYPE(Door1,Door)
  - TYPE(Door2,Door)
  - TYPE(Door3,Door)
  - TYPE(Door4,Door)
  - TYPE(Lightswitch1,Lightswitch)

- **INROOM(Box1,Room1)**
- **INROOM(Box2,Room1)**
- **INROOM(Box3,Room1)**
- **INROOM(Robot,Room1)**
- **INROOM(Lightswitch,Room1)**
- **PUSHABLE(Box1)**
- **PUSHABLE(Box2)**
- **PUSHABLE(Box3)**
- **ONFLOOR**
- **STATUS(Lightswitch1,Off)**
- **LOCINROOM(F,Room4)**

STRIPS E SHAKEY-3

- Operators:
  - **goto1(m)** – The robot goes to the coordinate location m
  - **goto2(m)** – The robot goes to the position next two item m
  - **pushto(m,n)** – The robot pushes object m next to item n
  - **turnonlight(m)** – The robot turns on lightswitch m
  - **climbonbox(m)** – The robot climbs onto box m
  - **climboffbox(m)** – The robot climbs off box m
  - **gothrudoor(k,l,m)** – The robot goes through door k from room l to room m

- In STRIPS we must specify for each operation what are the **preconditions** and what are the **effects**.

Disadvantages of Hierarchical Organizational Architectures

- While hierarchies have advantages, relying on a *world model* creates problems
  - Bottleneck on processing, particularly for control
    - Alternative is to create layers or hierarchies within the world model to match other subsystems (Kaebling, Simmons)
  - World model requires extensive representation which leads to two major problems:
    - Operates under the *closed world assumption*
    - *Frame problem*
- In practice, implementations are planning-centric

Adapted from R.Murphy

STRIPS OUTLINE

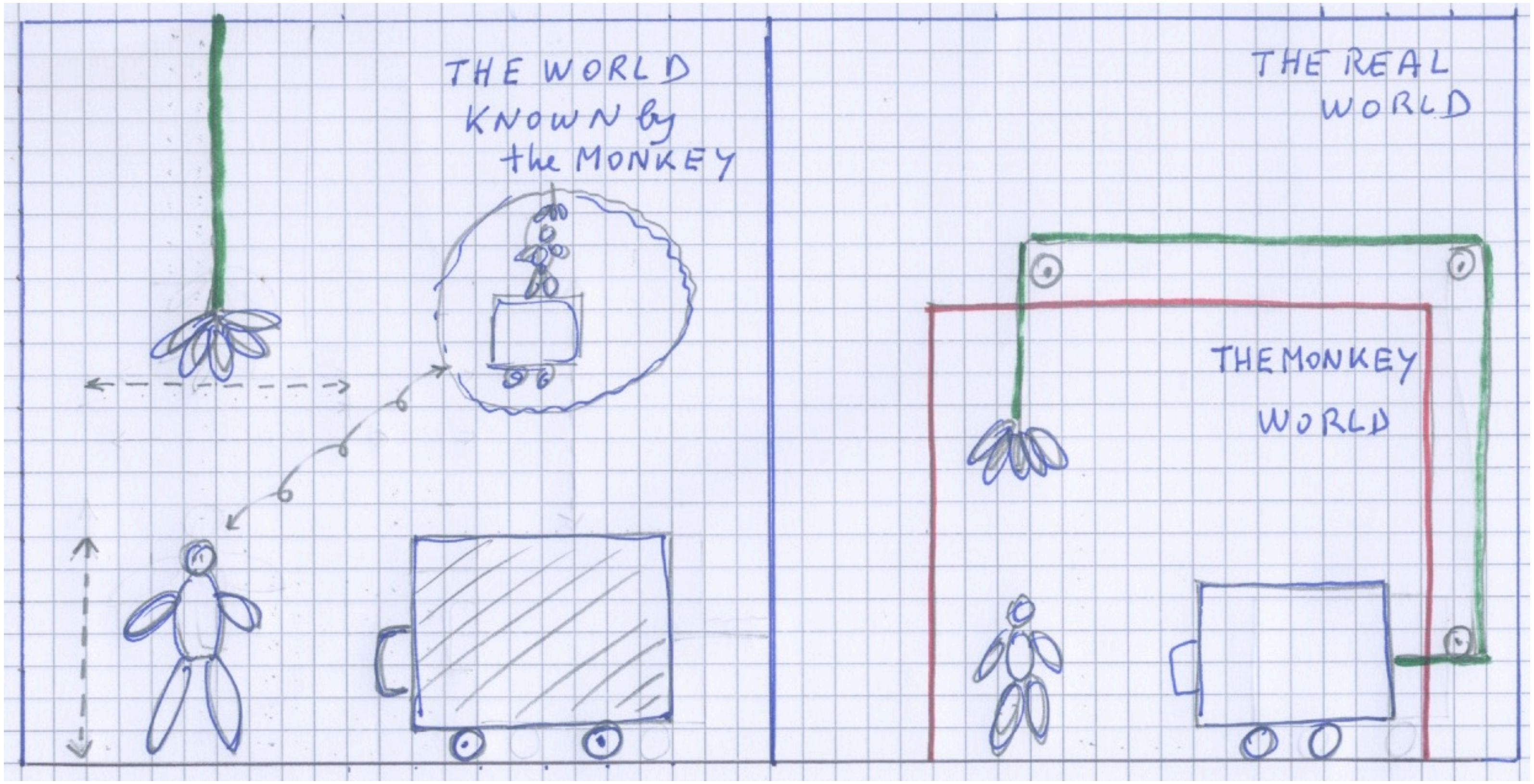
The programmer must implement:

- a representation of the model of the world
- a difference table with operators, preconditions and postconditions (lists of information to be added and removed)
- a difference evaluation function

- Strips assumes a ***closed world***
- Strips suffers a ***frame problem***

- ***Problem: not everything intelligent requires "deciding" (ex., reflexes or muscle memory such as riding a bike or driving a car)***

The Monkey and Banana Problem



- Pat Hayes introduced this nice gag in Brighton, at AISB-1976, the first Conference of the British Association of AI, that was also the 1st European Conference on Artificial Intelligence

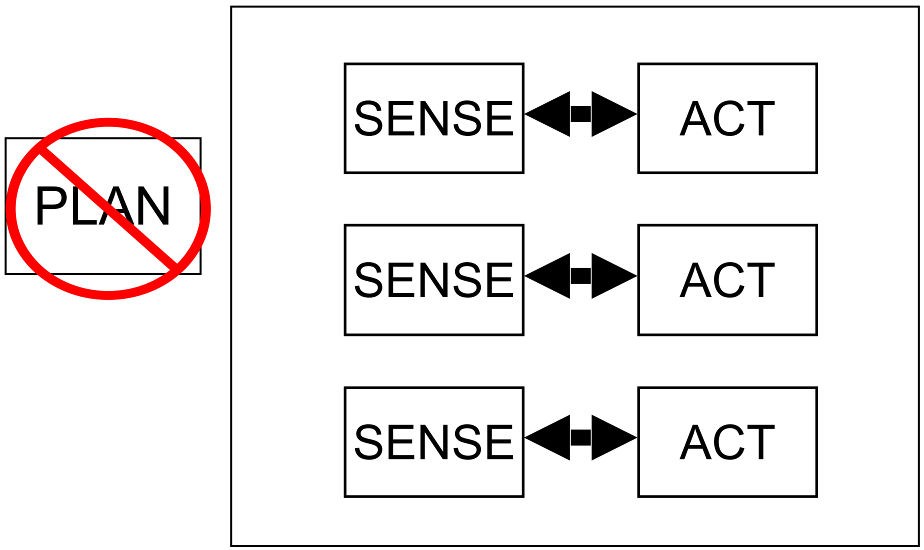
STRIPS OUTLINE

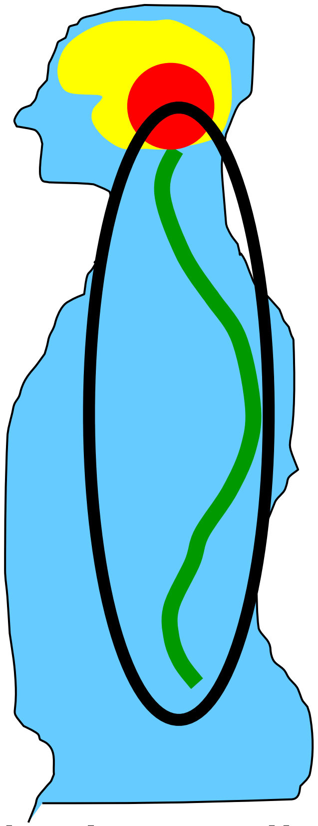
- The assumption of a **closed-world** and **the frame problem** make robots not reliable in the **real world**

- ***Moreover: not everything intelligent requires "deciding" (ex., reflexes or muscle memory such as riding a bike or driving a car)***

REACTIVE PARADIGM

Reactive (1986)



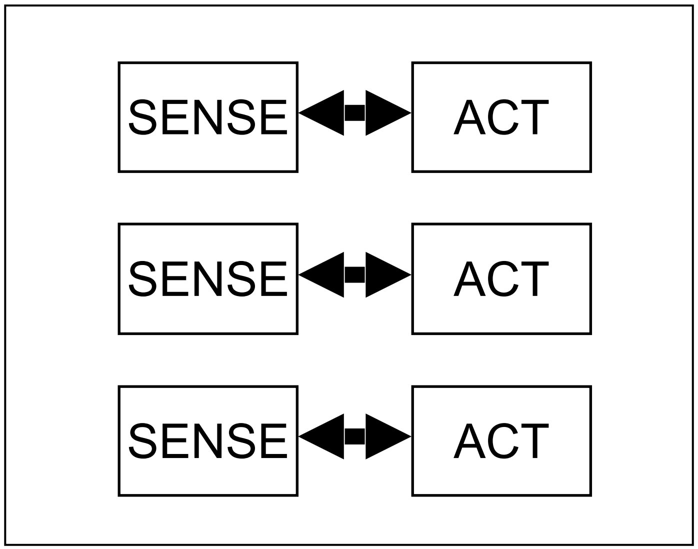


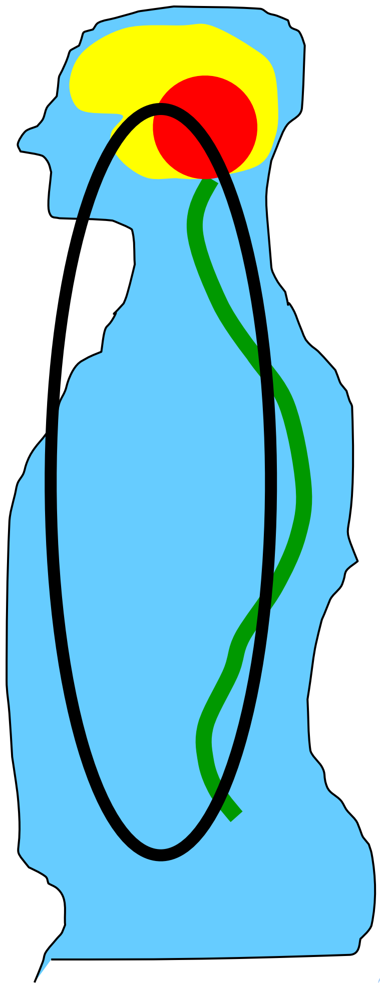
It arises from two reasons:

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

Adapted from R.Murphy

Reactive (1986)





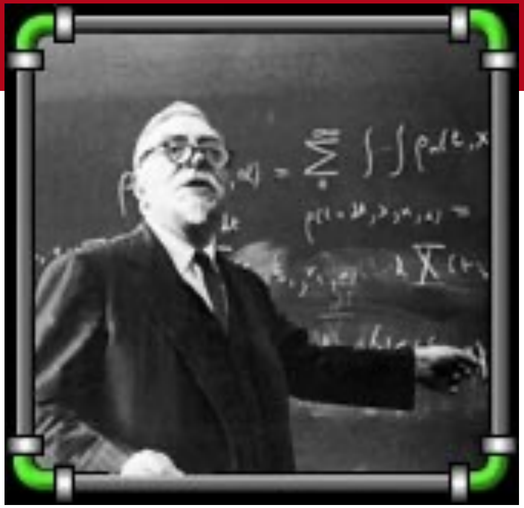
Users loved it because it worked

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

Slide di M. Reggiani

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.

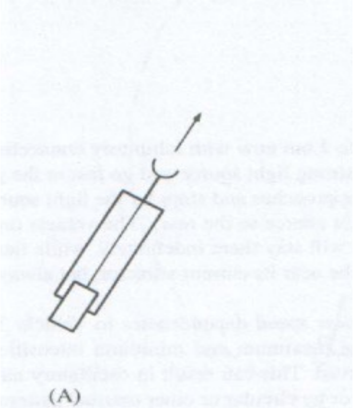
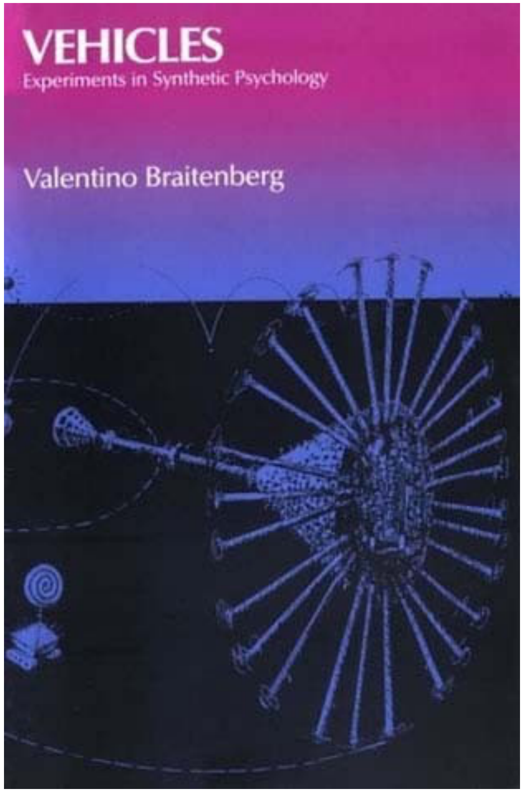
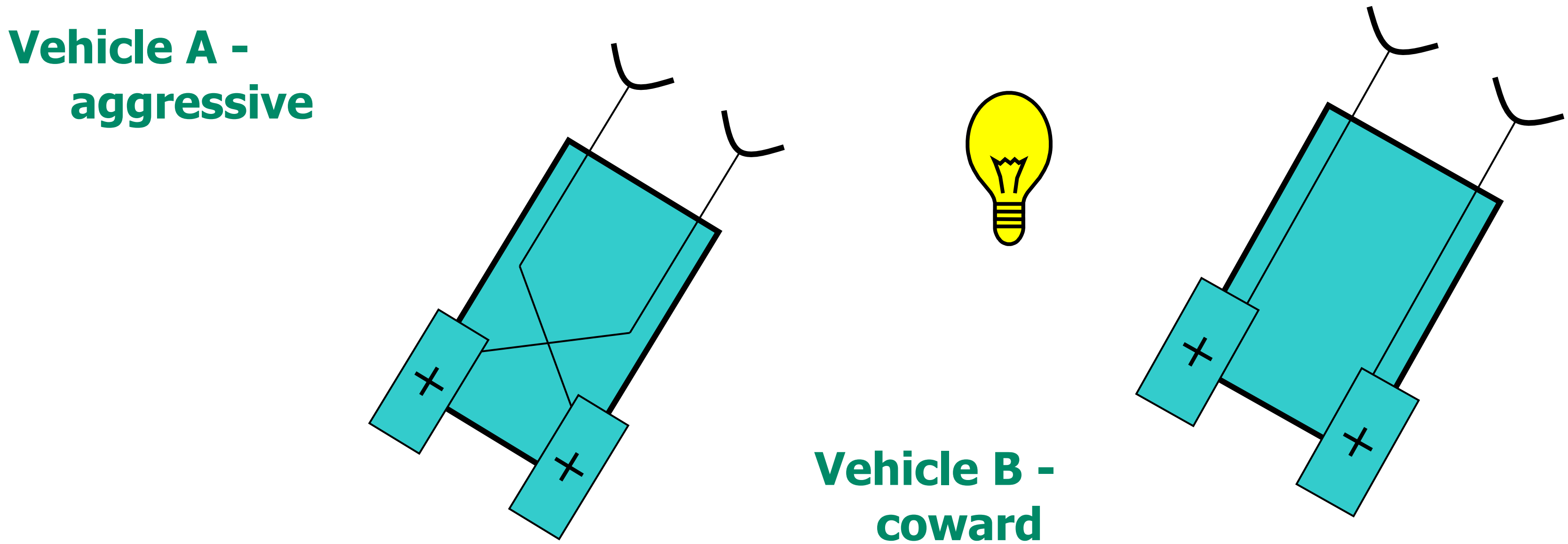


Figure 1.6
Braitenberg Vehicles



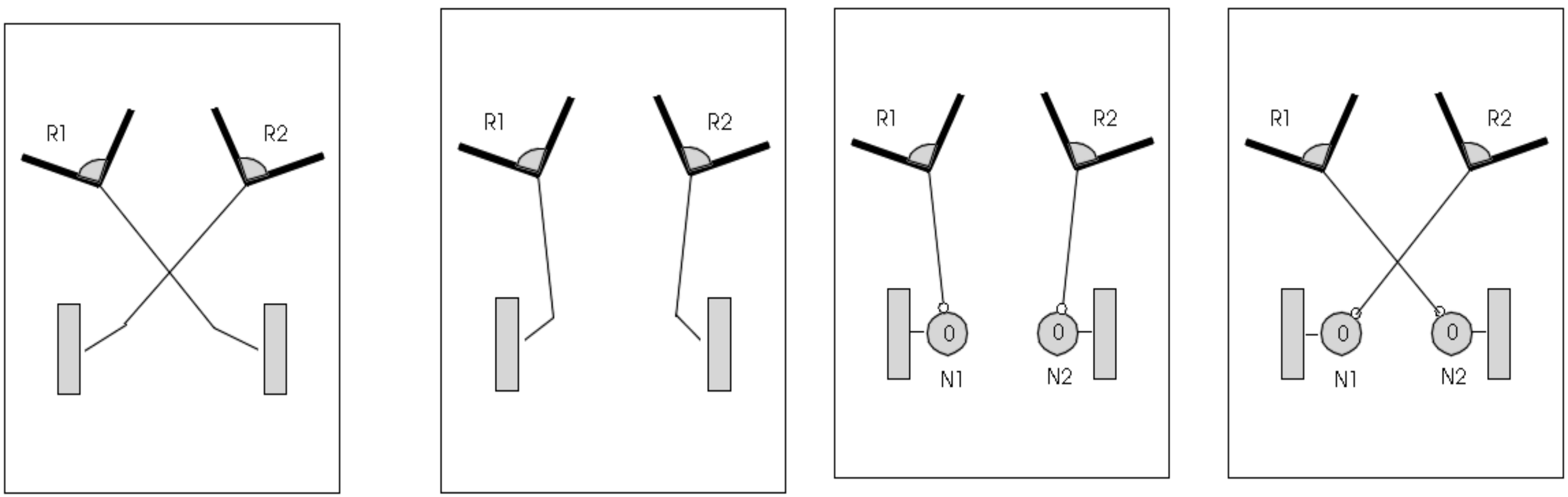
Each vehicle has sensors and motors. Vehicle No. 1 has one sensor and one motor. The more the sensor augment its input, the faster the motor goes.

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



- 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.