based on Gianni A. Di Caro lecture on

ROBOT CONTROL RCHITECTURES SINGLE AND MULTI-ROBOT SYSTEMS: A CASE STUDY IN SWARM ROBOTICS

POLITECNICO MILANO 1863

Cognitive Robotics

Robot Cognitive Architectures

Matteo Matteucci
matteo.matteucci@polimi.it

Artificial Intelligence and Robotics Lab - Politecnico di Milano

Dealing with the real world is hard!

Providing autonomy to robots and vehicles can be a cumbersome matter:

- Complex tasks (e.g., autonomous driving from Kolkata to New Delhi)
- Multiple issues to take care of (e.g., batteries, avoid obstacles, don't fall to stairs)
- Sensor/effector noise and uncertainty
- Dynamic environment and unexpected events
- Lack of precise information / models
- Mechanical constraints (e.g., non-holonomic constraints)

To face such a complexity we need to define and organize a set of building blocks / tools to take informed decisions ➔ *Robot Cognitive Architecture*

Robot architectures

- A principled way of organizing a control system. In addition to providing structure, it also imposes constraints on the way the control problem can be solved (M. Mataric)
- The description of a set of architectural components and how they interact (Dean and Wellman)

Different architectural approaches (based on the same building blocks) produces different designs/results of the same general concept …




Robot cognitive paradigms

Classical cognitive approaches are based on different “mental models” (*paradigms)*

Paradigm:
A philosophy or set of assumptions and/or techniques which characterize an approach to a class of problems (R. Murphy).

In the case of robots, it defines the general model of operations.

In classic robot cognitive architectures three main paradigms exists

- *Deliberative / Model-based / Hierarchical / Orizontal*
- *Reactive / Behavioral / Vertical*
- *Hybrid / Multi layered (Deliberative + Reactive)*

Deliberative paradigm: “think hard, act later”

Deliberation:
Thoughtfulness in decision and action ➔ Thinking hard

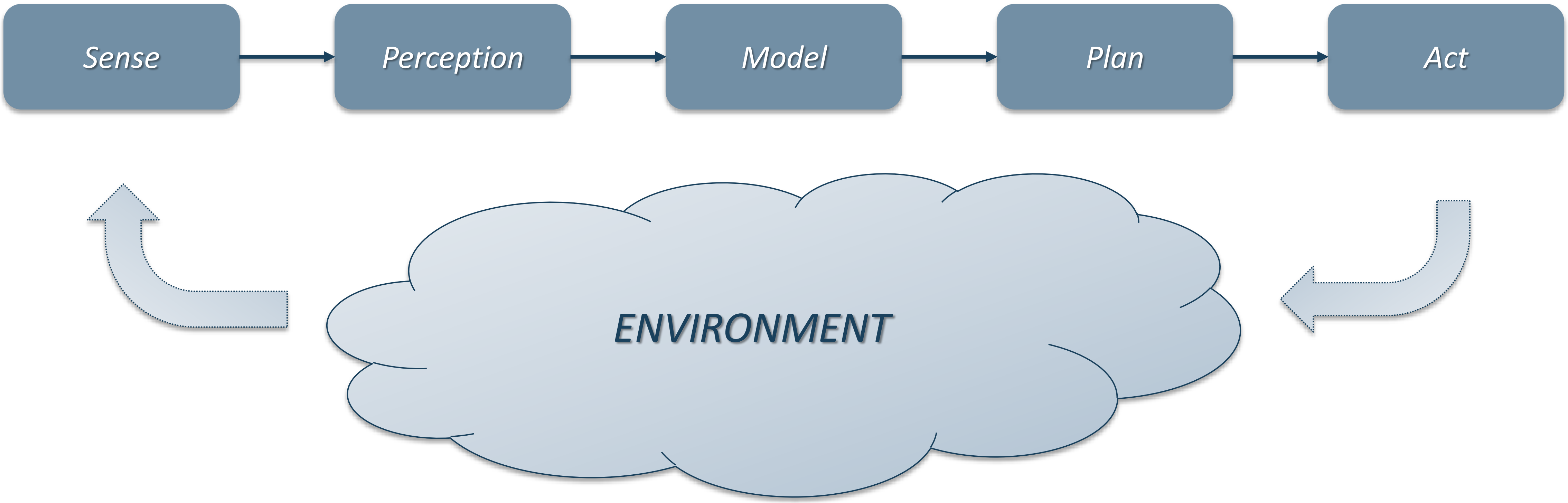
Deliberative control grew out of ***Classical AI*** (60’s - 80’s) and its vision of human intelligence. To be intelligent, machines/robots have to be able to perform some intensive forms of “thinking”, that in turn would require:

- *Internal models of the world*
- *Search through possible solutions*
- *Planning and reasoning to solve problems*
- *Hierarchical system organization*

Top-Down Approach to Problem Solving

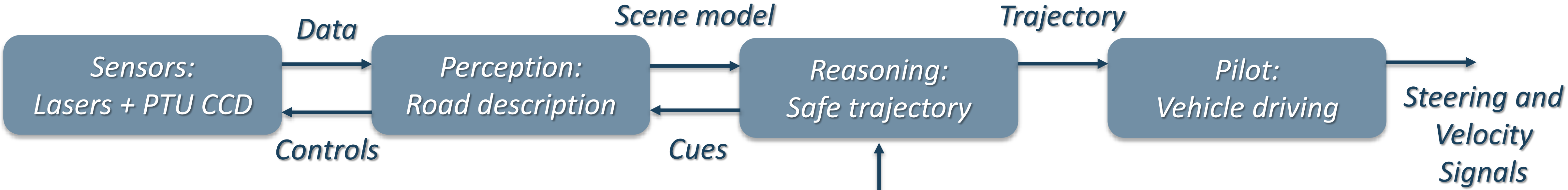
Deliberative paradigm: “think hard, act later”

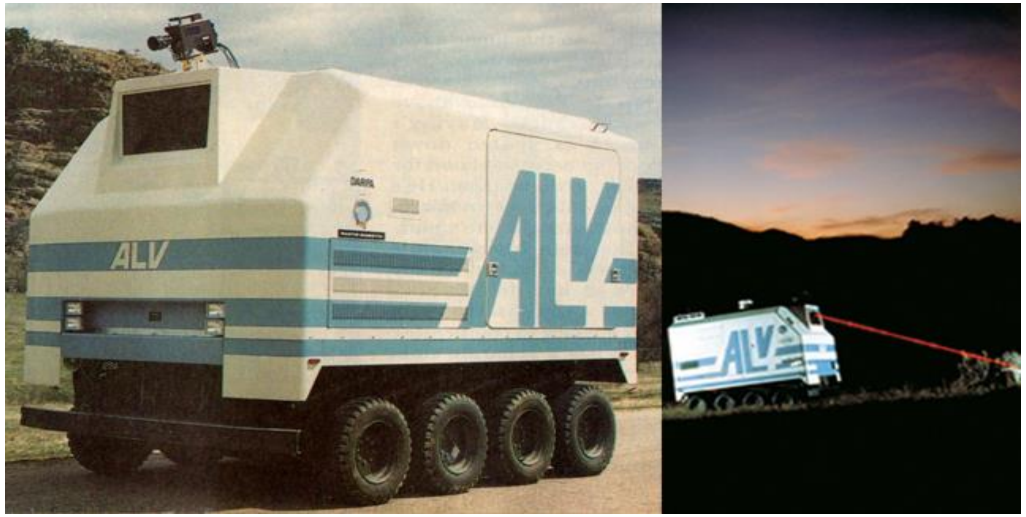
The deliberative paradigm adopt a well defined pipeline of functional modules



Example of deliberative architecture: *ALVIN*

The *Autonomous Land Vehicle* (ALV) Alvin (CMU mid to end 80’s) was the first on-road and off-road vehicle performing autonomous navigation.



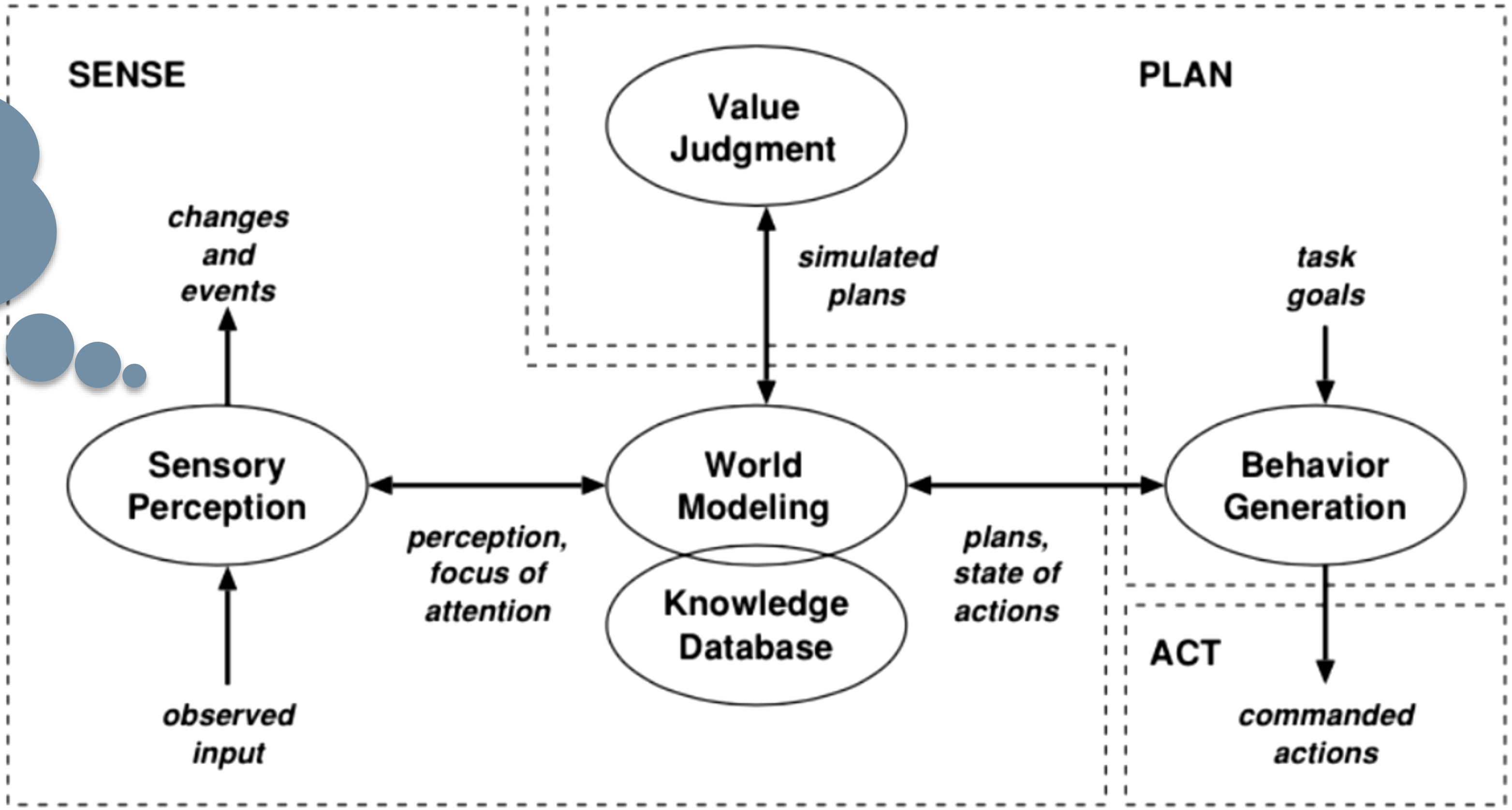


Check SRI Shakey and Flakey and their STRIPS planner!

A more general example: *RCS*

Real-time Control System (RCS) Architecture was proposed by J. Albus (NIST 1986--) as a flexible architecture for manufacturing robots

Semi-autonomous control: human provides world model, decides mission, decomposes it into a plan …



Source: R. Murphy, *AI Robotics*, MIT Press, 2000

Drawbacks of deliberative architectures (time scale)

Time scale issues

- Inability to react rapidly (e.g., in case of emergency the robot must still sense + model + plan before acting) and to meet multiple goals

- The planning step can potentially be very expensive/long in large state spaces (sensor + model states)
  - This might require the robot to stop and wait for the next plan
  - A large planning time compared to robot speeds “encourages” open loop control, to avoid keep doing (expensive) re-planning. However, this might be a very bad idea in dynamic or uncertain environments

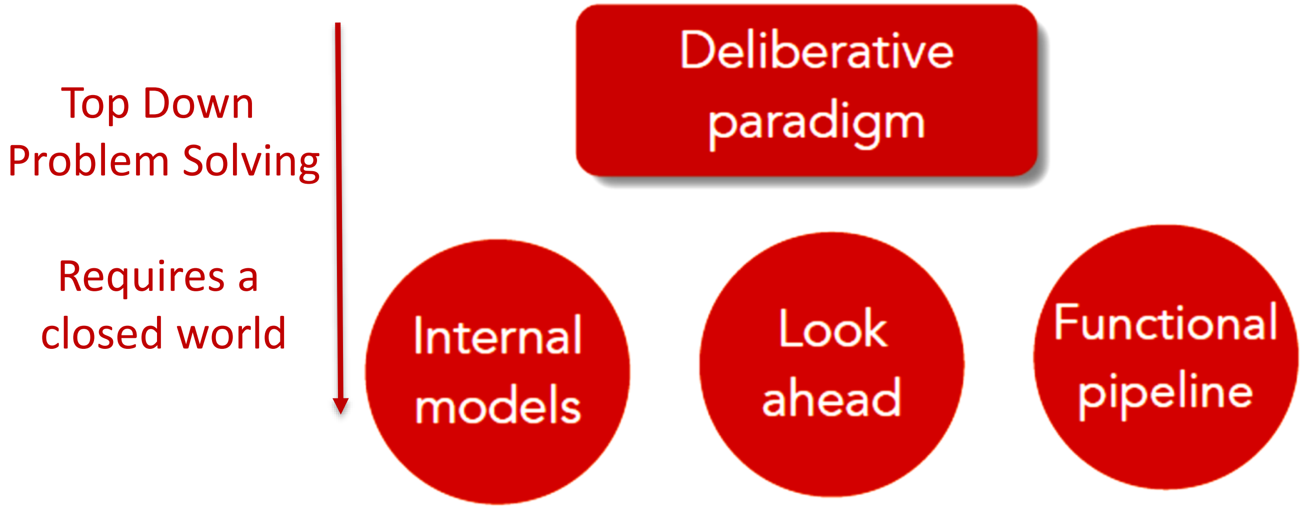
Drawbacks of deliberative architectures (information)

Information issues

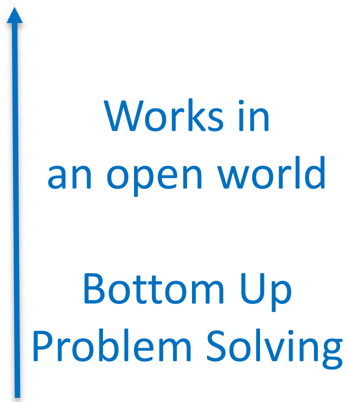
- The representation of the state space needs to be accurate, comprehensive, and up to date. This is not always the case in the real-world, and requires continual updating of the world model (which takes time … see previous issues).
- The robot needs to know with precision the state of the world and of the plan execution at all times. What about dynamic environments? Effector errors? …

Closed-world assumption

Reactive paradigm: “don’t think, react!”



Starting from the mid 1980s, a number of different views (mostly *bio-inspired*) and approaches were developed and employed in robotics, and in AI, moving from symbolic to sub-symbolic / neural models



Reactive paradigm: “don’t think, react!”

Ethological view (Behavior):
Direct mapping of sensory inputs to a pattern of motor actions that are then used to achieve a task.

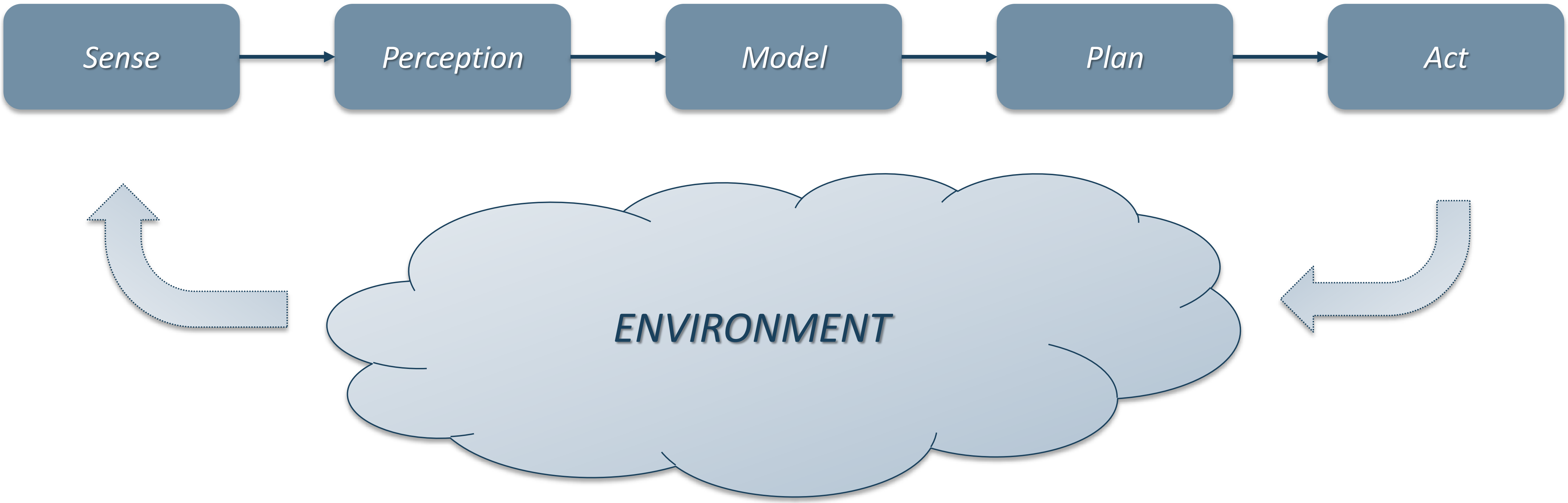
Mathematical view (Function):
A transfer function, transforming sensory inputs into actuator commands

Main difference with respect to the deliberative approach

- Concurrent mode vs. Sequential mode
- Vertical decomposition vs. Horizontal decomposition (alternative view)

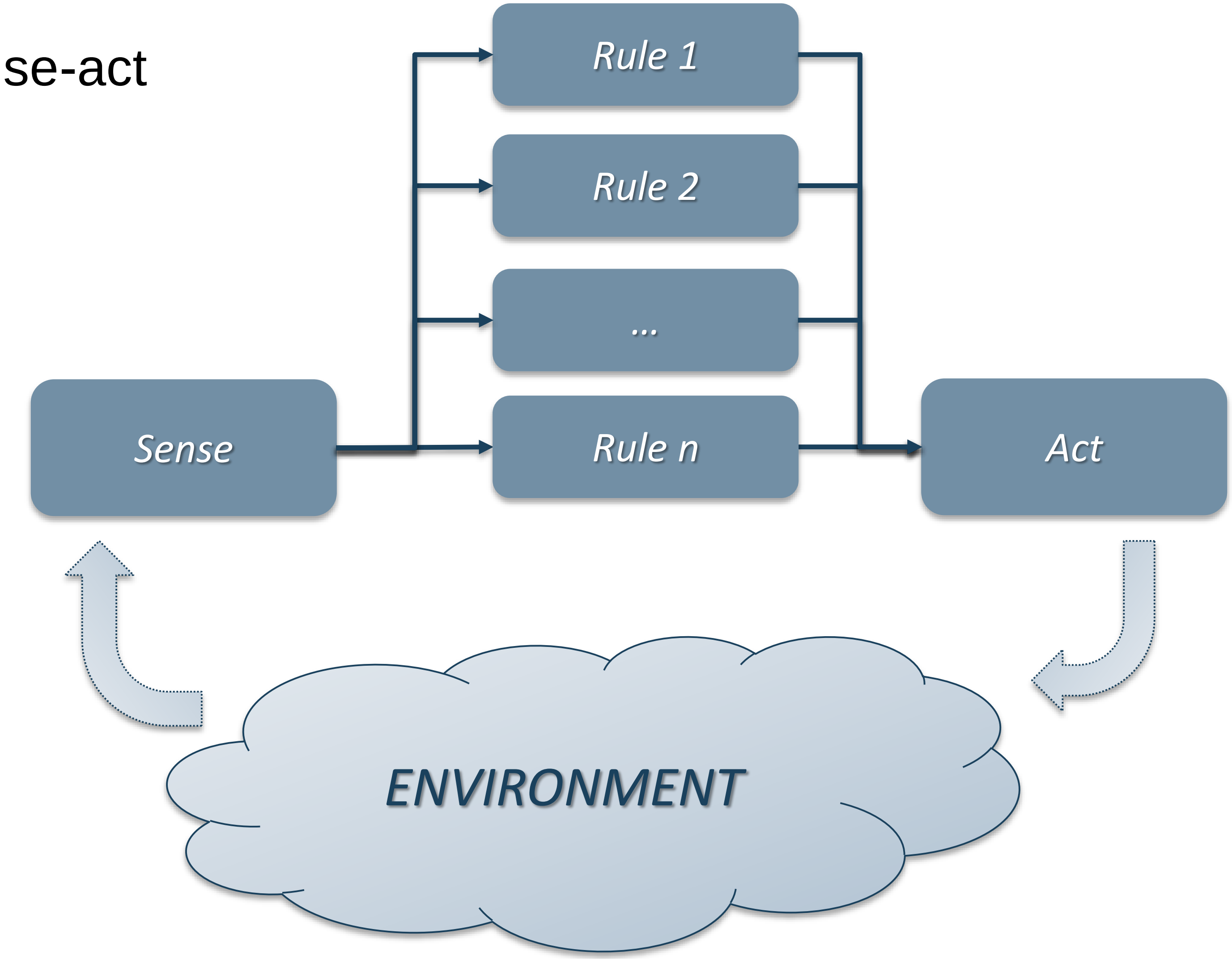
Deliberative paradigm: “think hard, act later”

The deliberative paradigm adopt a well defined pipeline of functional modules



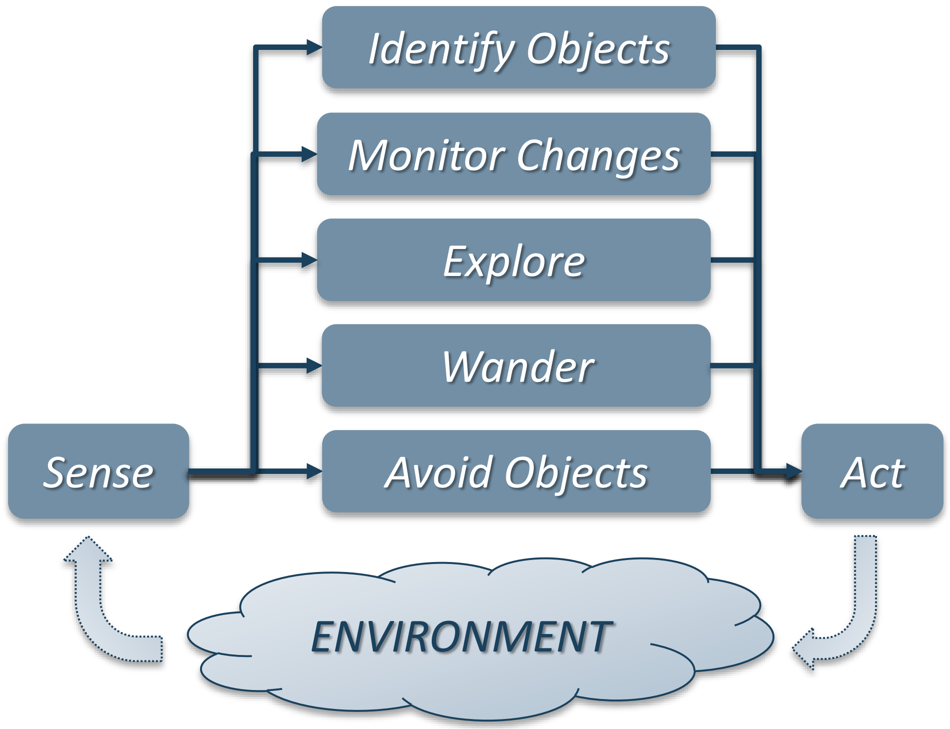
Reactive paradigm: “don’t think, react!”

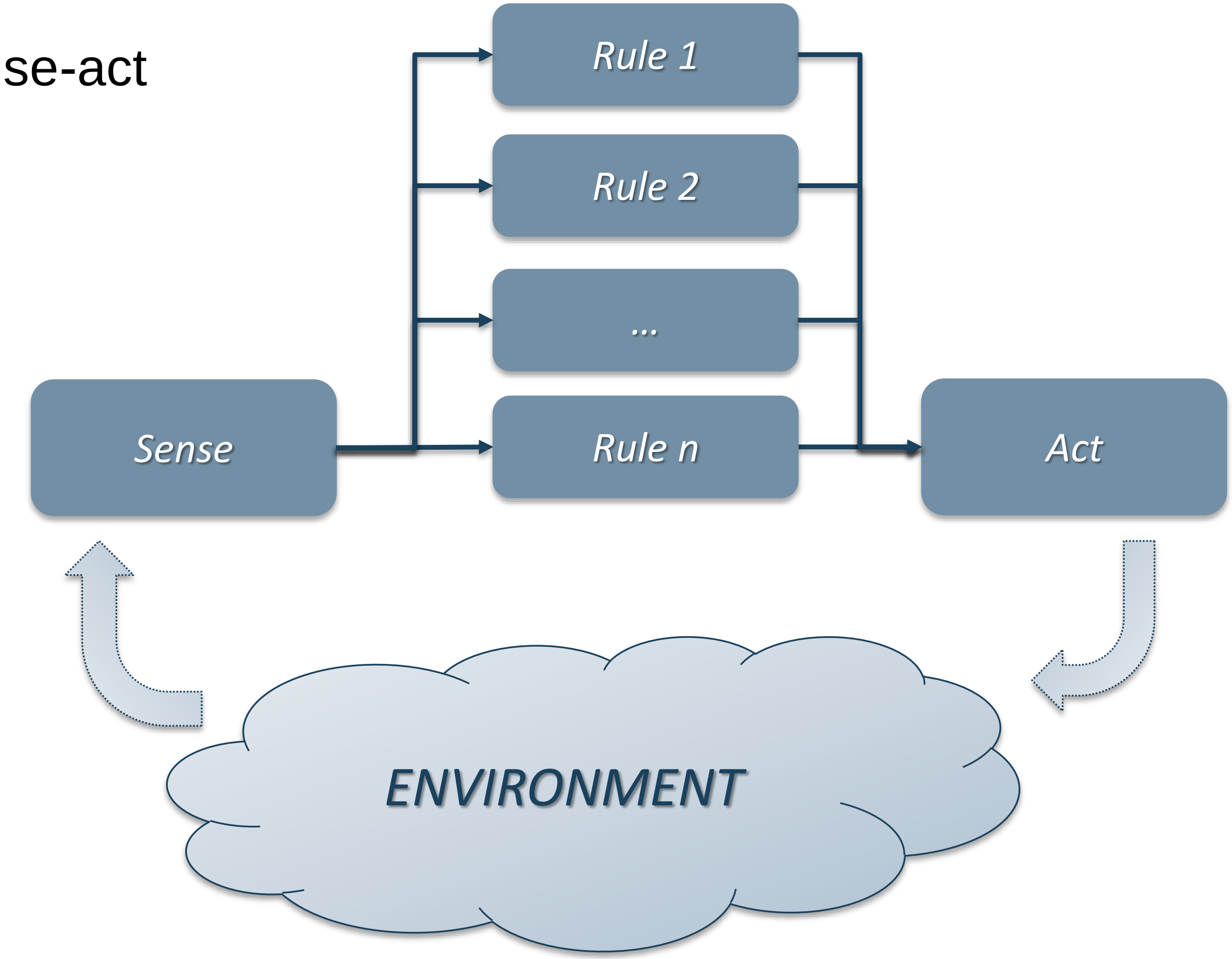
The Reactive paradigm executes sense-act transfer rules behaviors



Reactive paradigm: "don't think, react!"

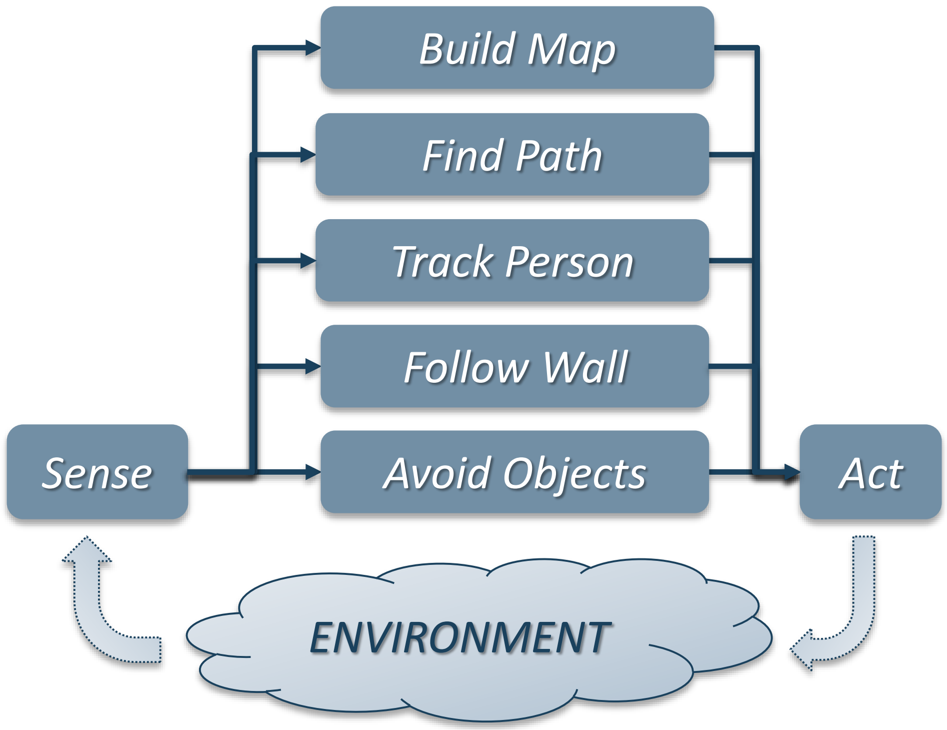
The Reactive paradigm executes sense-act transfer rules behaviors

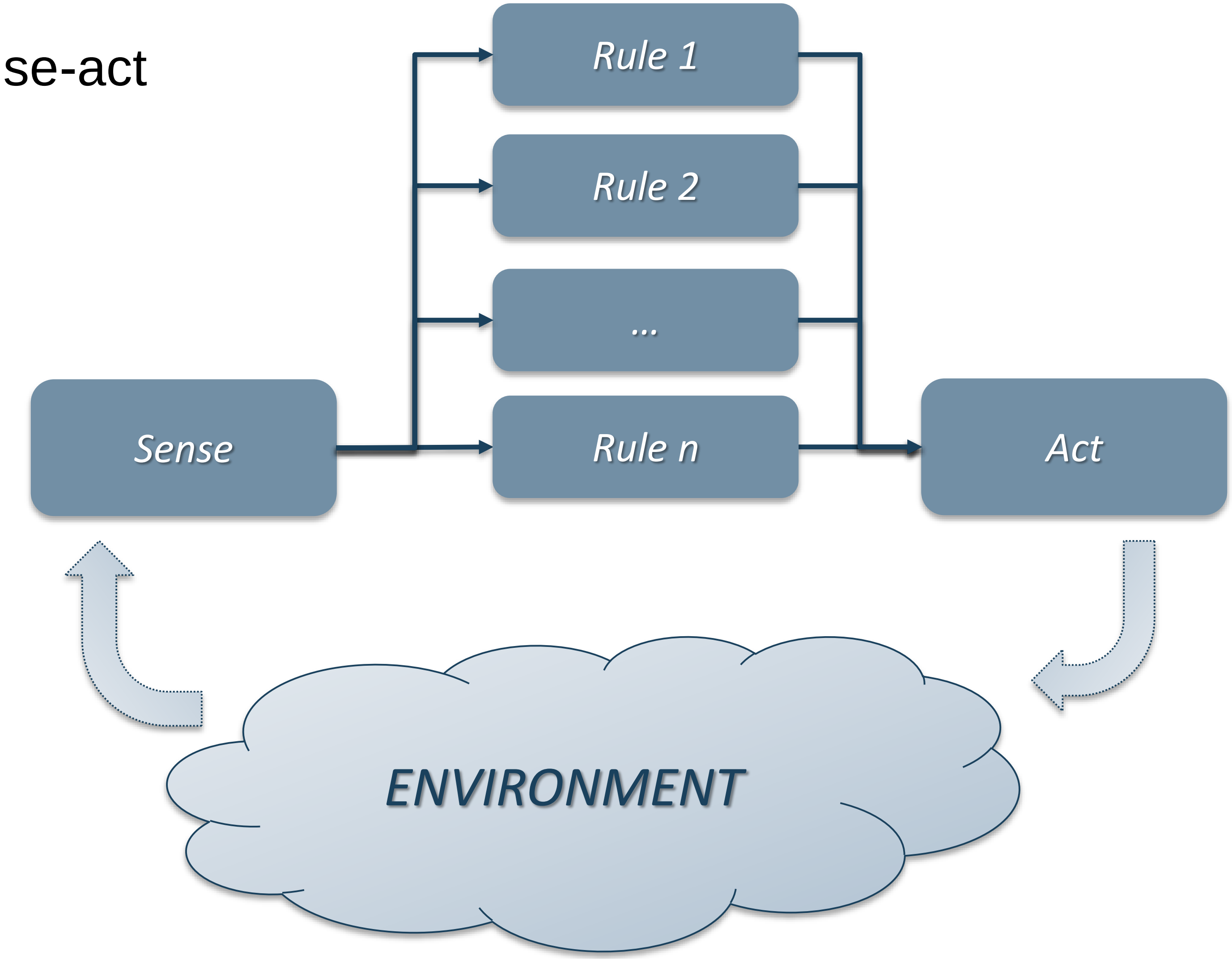




Reactive paradigm: “don’t think, react!”

The Reactive paradigm executes sense-act transfer rules behaviors

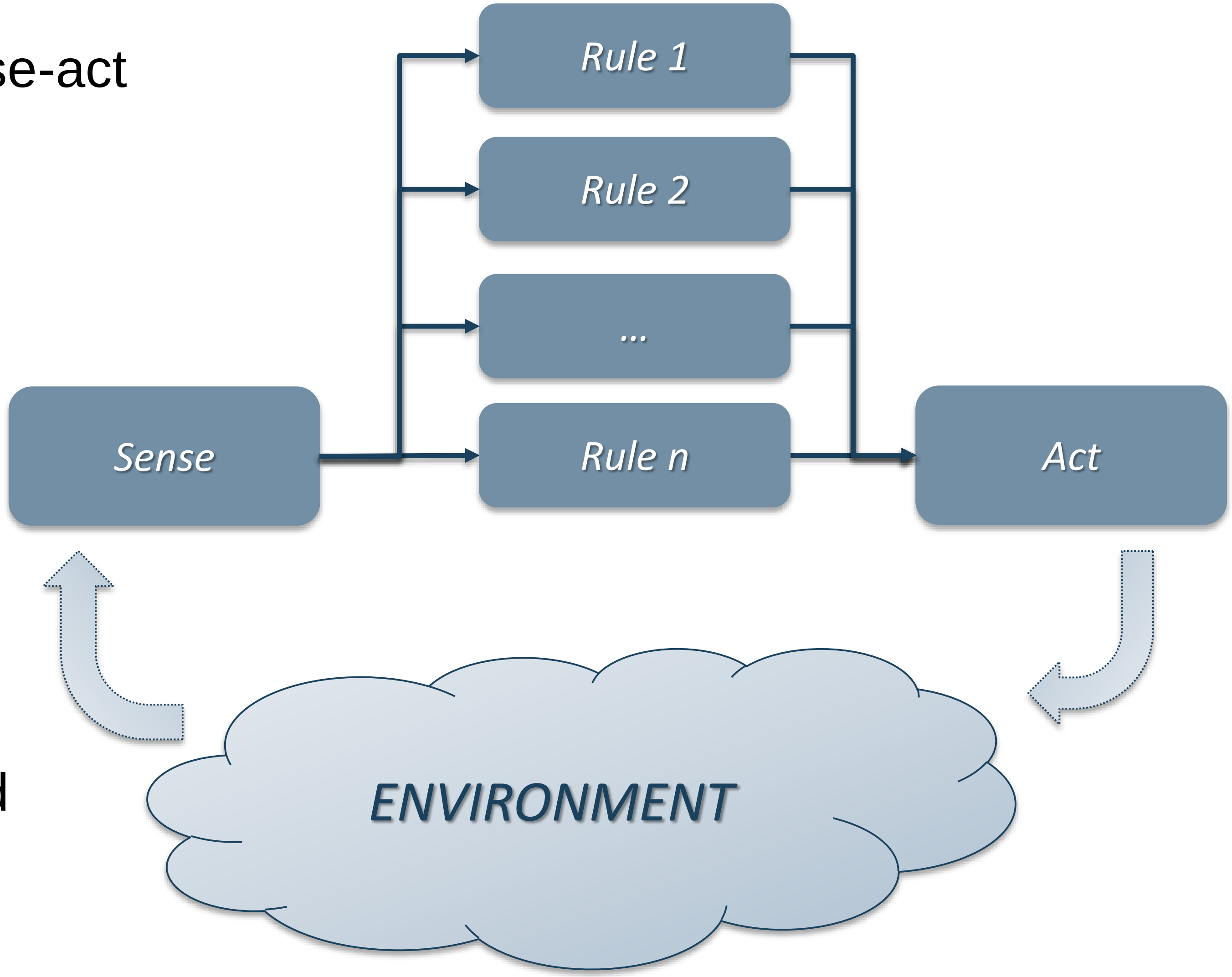




Reactive paradigm: “don’t think, react!”

The Reactive paradigm executes sense-act transfer rules behaviors

- Where the Reactive paradigm finds its roots?
- What is the exact nature/characteristic of the Sense-Act rules?
- How the Act output from the different rules is arbitrated as a single, coherent command to the effectors?



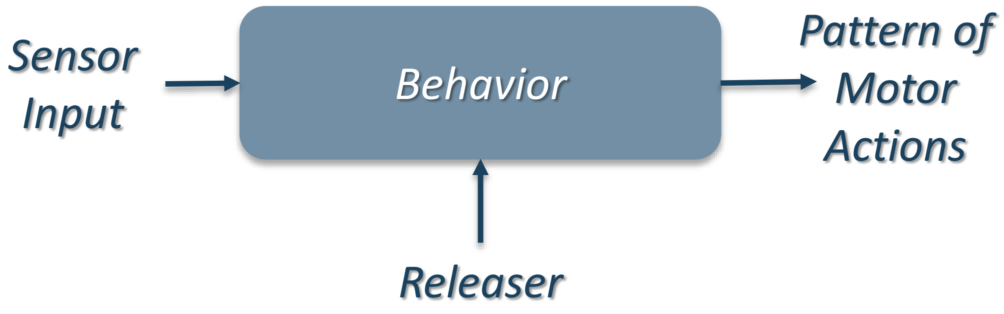
The biological roots of the reactive approach

Dissatisfaction with the limitations of the Deliberative approach led to observing that:

- Animals live in an open world, and roboticists would like to overcome the closed world assumption
- Many “simple” animals exhibit individual and collective intelligent behavior yet have virtually no brain. Therefore, they must be doing something to manage world’s representation complexity!

Reactive rules as *Behaviors*

A fundamental building block of natural intelligence is a behavior: a mapping of sensory inputs to a pattern of motor actions, which then are used to achieve a task



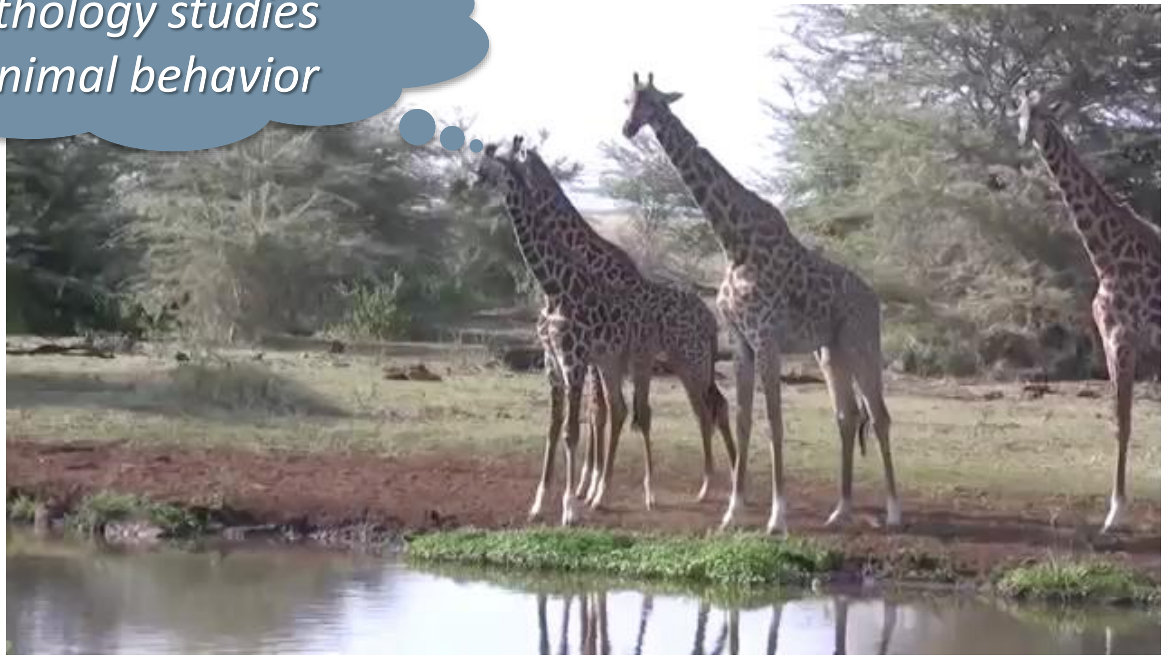
Reactive rules as *Behaviors*

A fundamental building block of natural intelligence is a behavior: a mapping of sensory inputs to a pattern of motor actions, which then are used to achieve a task

Sensor input: `Water source detected`
Releaser: `Giraffe is thirsty` `No predators`
Action pattern:

```
Move head checking for predators
Put legs in right position,
Lower the neck
Adjust legs position
Drink rapidly
Neck up and check surroundings
```

Ethology studies animal behavior



What kind of (animal) behaviors?

Increasing level of complexity

Reflexive behaviors

- Stimulus-response, “hardwired” behaviors.
- Stimulus is directly connected to the motor action to produce the fastest response time.
- No cognition: if you sense it, you do it!

Reactive behaviors:

- Learned, and then consolidated so they can be executed without conscious thought, but can be changed by conscious thought.

Conscious behaviors:

- Deliberative, requiring conscious thought, possibly combining previously developed behaviors

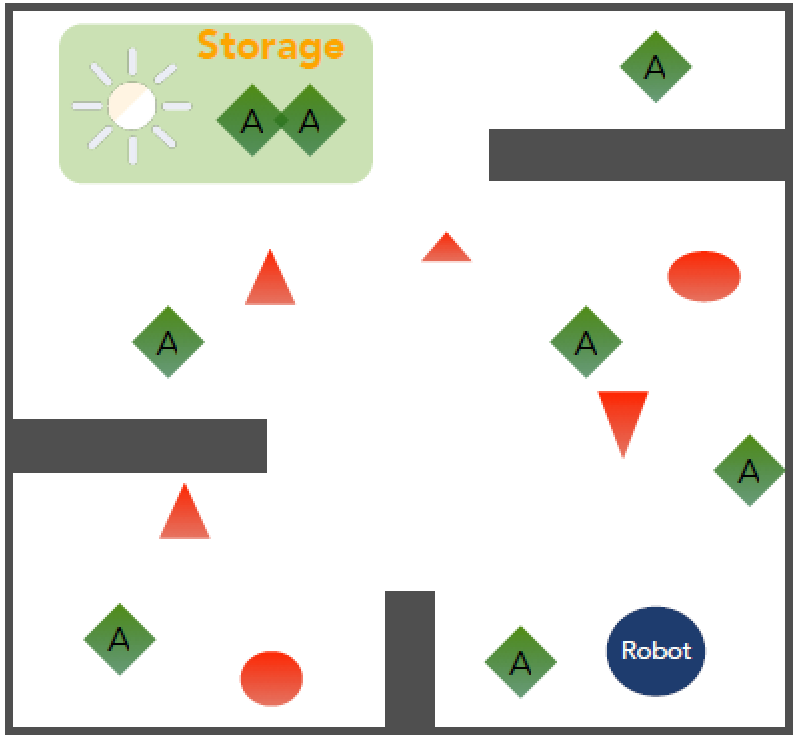
A practical example: object collection

Task specification

- Search for type A objects
- When an A object is found, brought it (pushing) at a storing location identified by a bright light
- Collect as many A objects as possible
- Other objects cannot be pushed
- The environment can feature walls

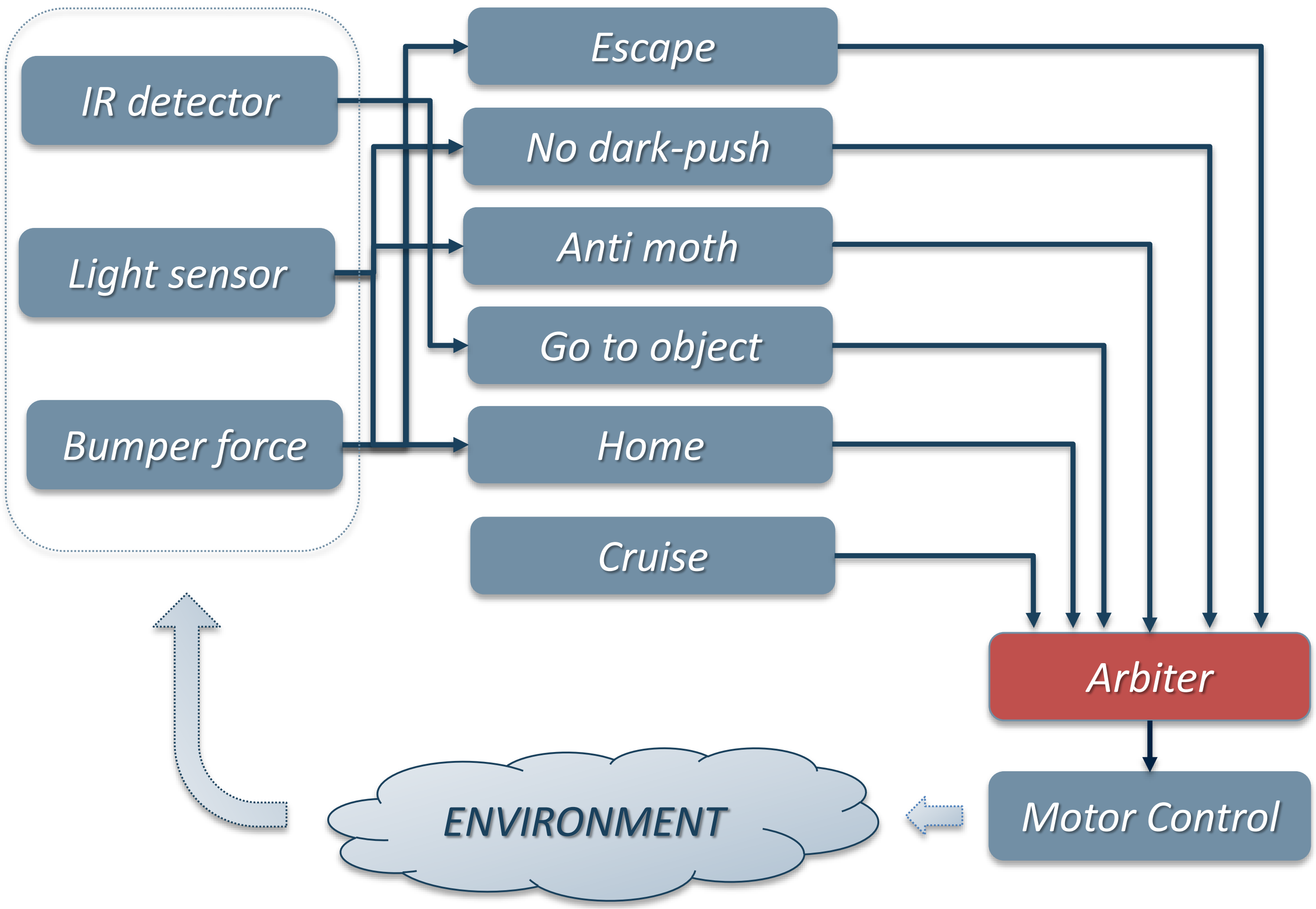
The robot:

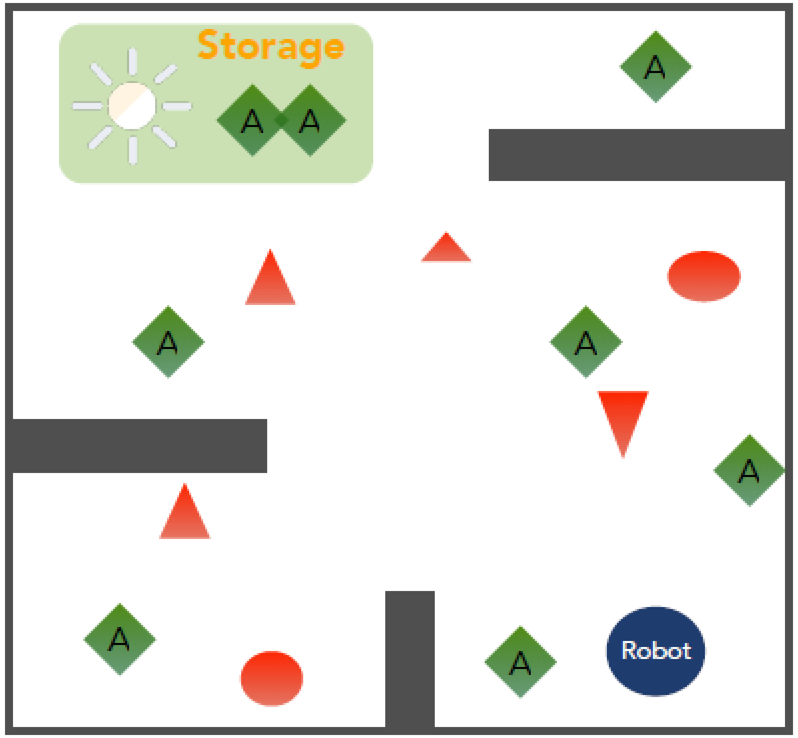
- Frontal IR emitters / detectors
- Light sensors
- Frontal bumper
- Two standard wheels



Seems an easy task …

A practical example: object collection



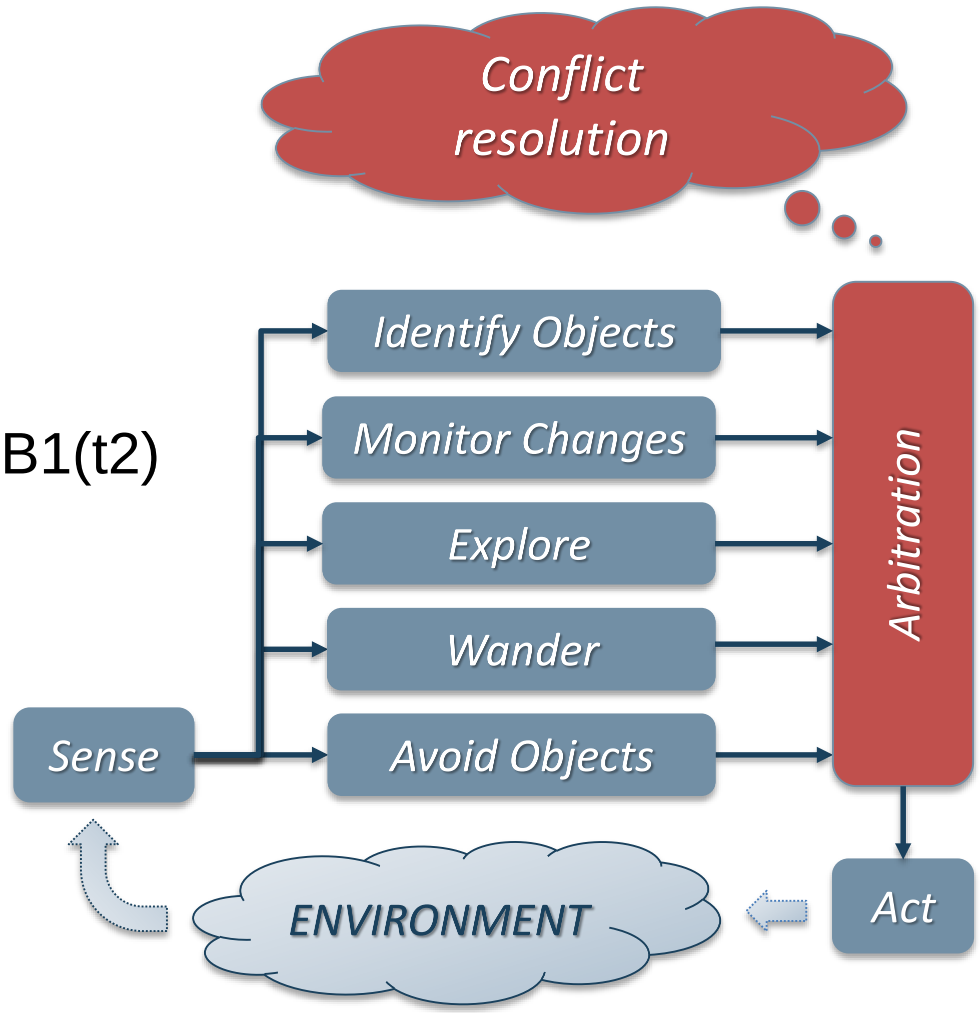


<u>Emergent behavior</u>: a set of simple behaviors that, when acting together, produce the overall desired activity

Behavior arbitration

Several aproaches have been proposed

- Fixed priority: B1(t) ≻ B2(t), ∀*t*
- Alternate: B2([t1,t2]), B1([t2,t3])
- Variable priority B1(t1) ≻ B2(t1), B2(t2) ≻ B1(t2)
- Subsumption
  - Suppression: $B^{New} \succ B^{old}$
  - Inhibition: $B^{New} \wedge B^{Old}$ then ∅
- Voting: {R1, R2, R3}: X, {R4}: Y, then X
- Averaging / Composition: B1 ⊕ B2

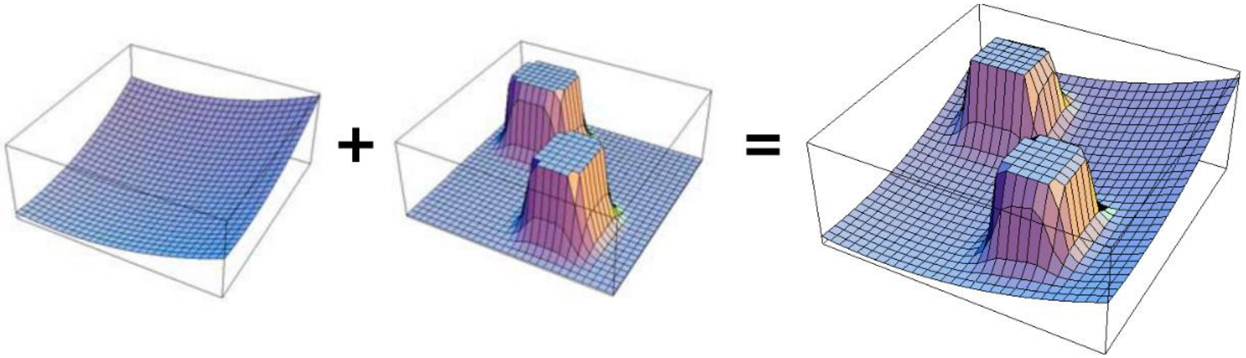


Composition approach by Potential Fields

A navigation method commonly adopted uses *Motor schemas / Potential fields.*

The robot can be represented as a particle under the influence of an *artificial potential field U(q)* which superimposes:

- Repulsive forces from obstacles
- Attractive force from goal(s)



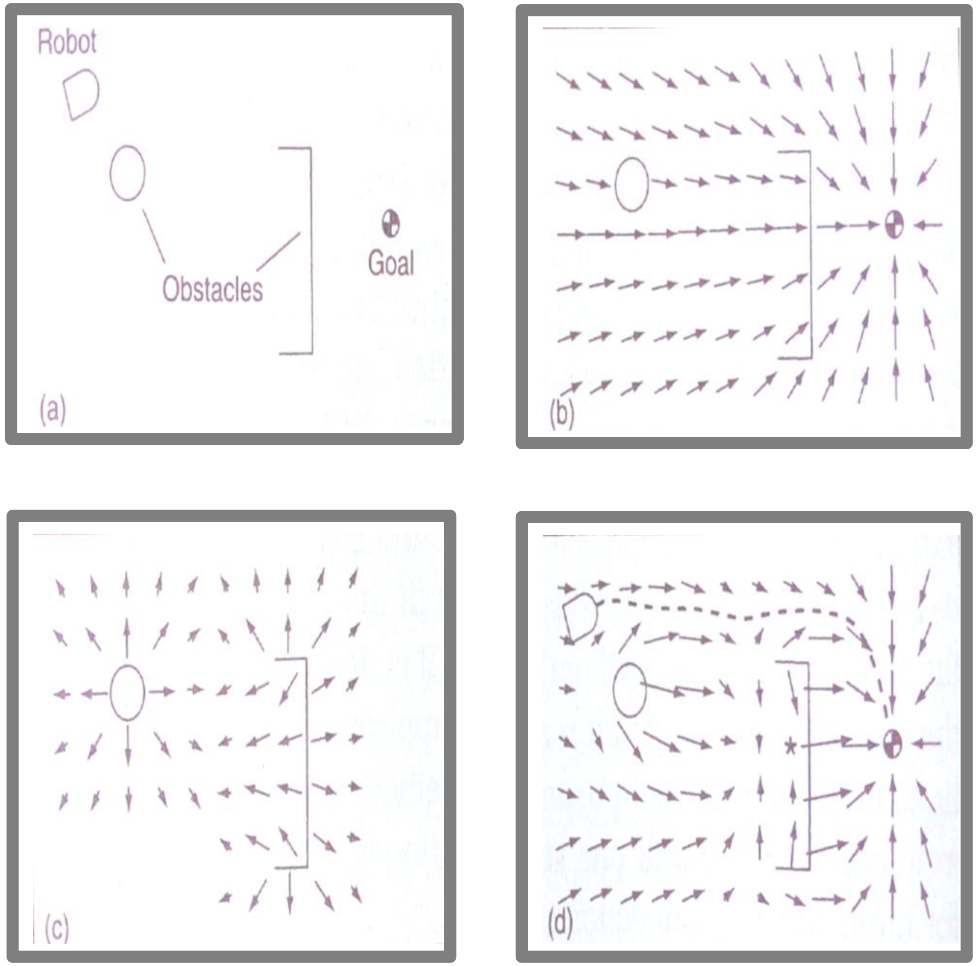
Different behaviors *feels* different fields, and the arbiter combines their proposed motion vectors

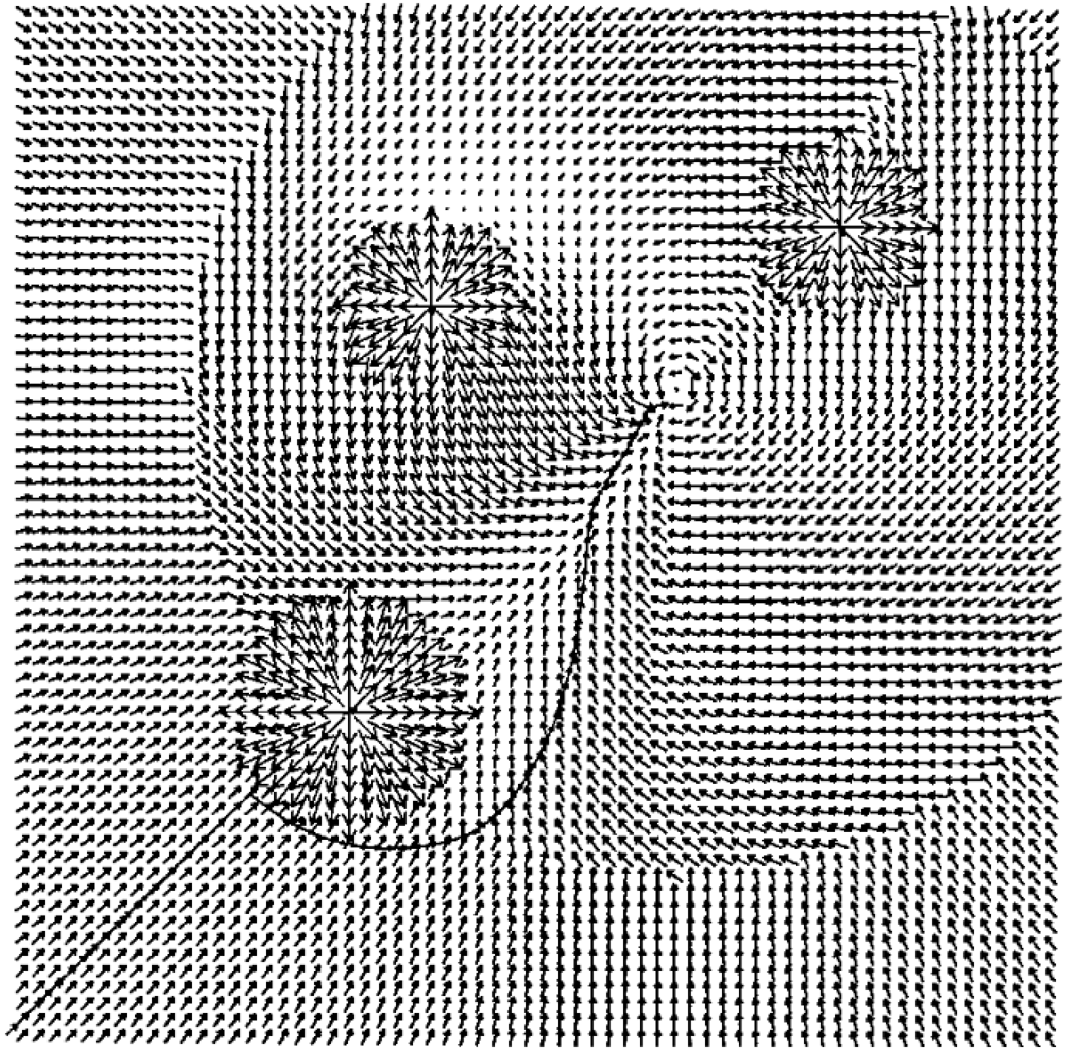
Following a gradient descent moves the robot towards the minima (goal = global minimum)

$$U(q) = U_{att}(q) + U_{rep}(q)$$

$$\vec{F}(q) = -\vec{\nabla}\, U(q)$$

Potential fields at work





Reactive paradigm and beyond

- Real-time capabilities, works in open worlds, doesn’t need models
- Task-oriented decomposition of the controller in parallel modules,
- Can deal with multiple sensors operating at different time-scales
- Easily extensible / modifiable by adding / removing modules/behaviors
- A network of behaviors can be created to build up more complex behaviors
- Rely only on minimal state, have no memory, no learning, no internal models

Bottom-up design, emergent behavior:

- Looks easier compared to top-down specification, but it’s also a sort of art
- Issues with predictability and formal analysis

High-level reasoning and planning:

- Planning, and more general, cognition and deliberation are useful, if not needed in complex scenarios

They complete each other

Hybrid architectures: Deliberative + Reactive

The most used approach today, but still an art!

Combine the best of two world in a single architecture

- Deliberative:
  - Representations
  - Models
  - Planning

Strategic planning / reasoning

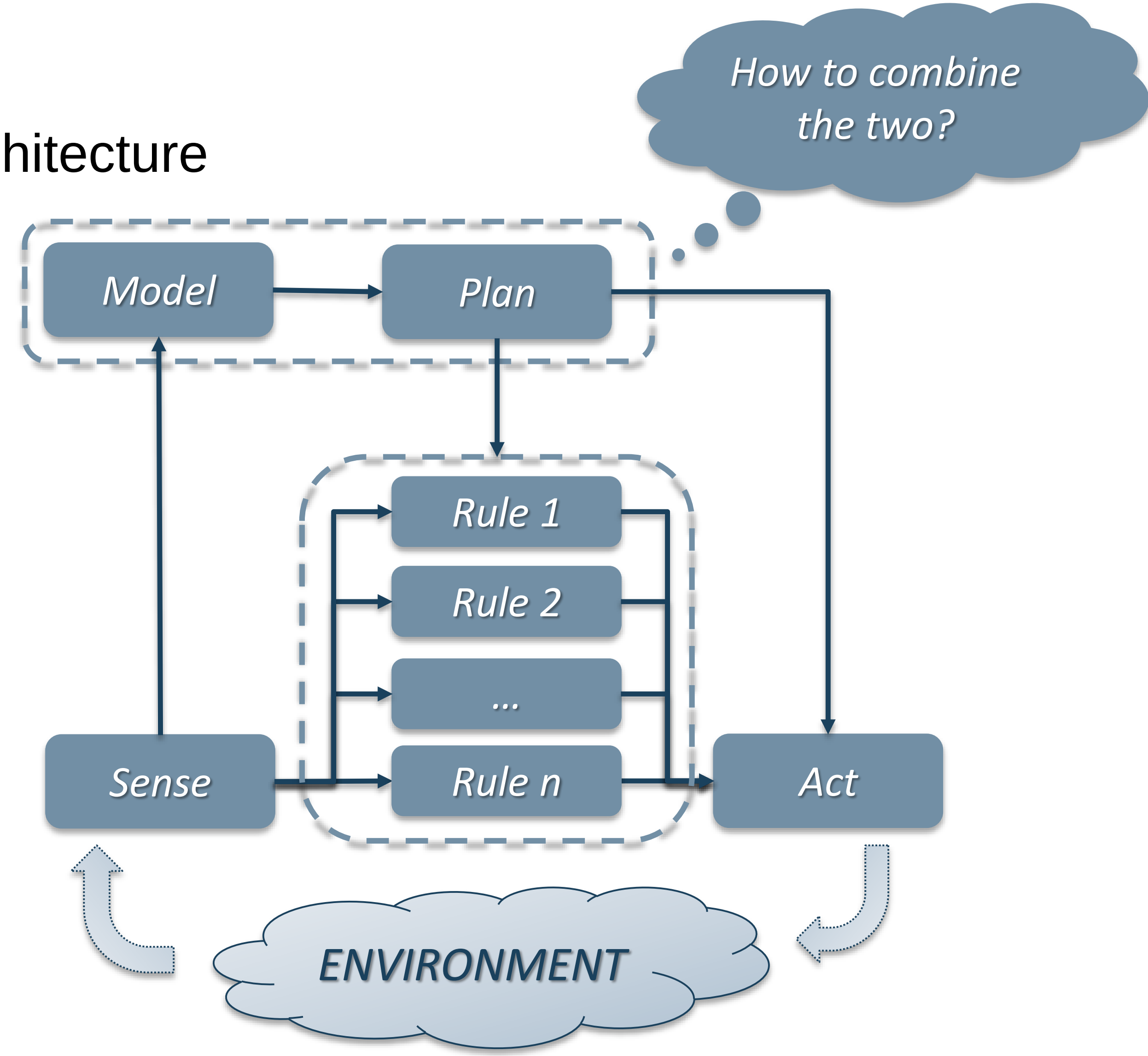
- Reactive
  - Real-time
  - Multiple goals
  - Robustness
  - Flexibility
  - Modularity

Low(er)-level controls and behaviors

Hybrid architectures: Deliberative + Reactive

Combine the best of two world in a single architecture

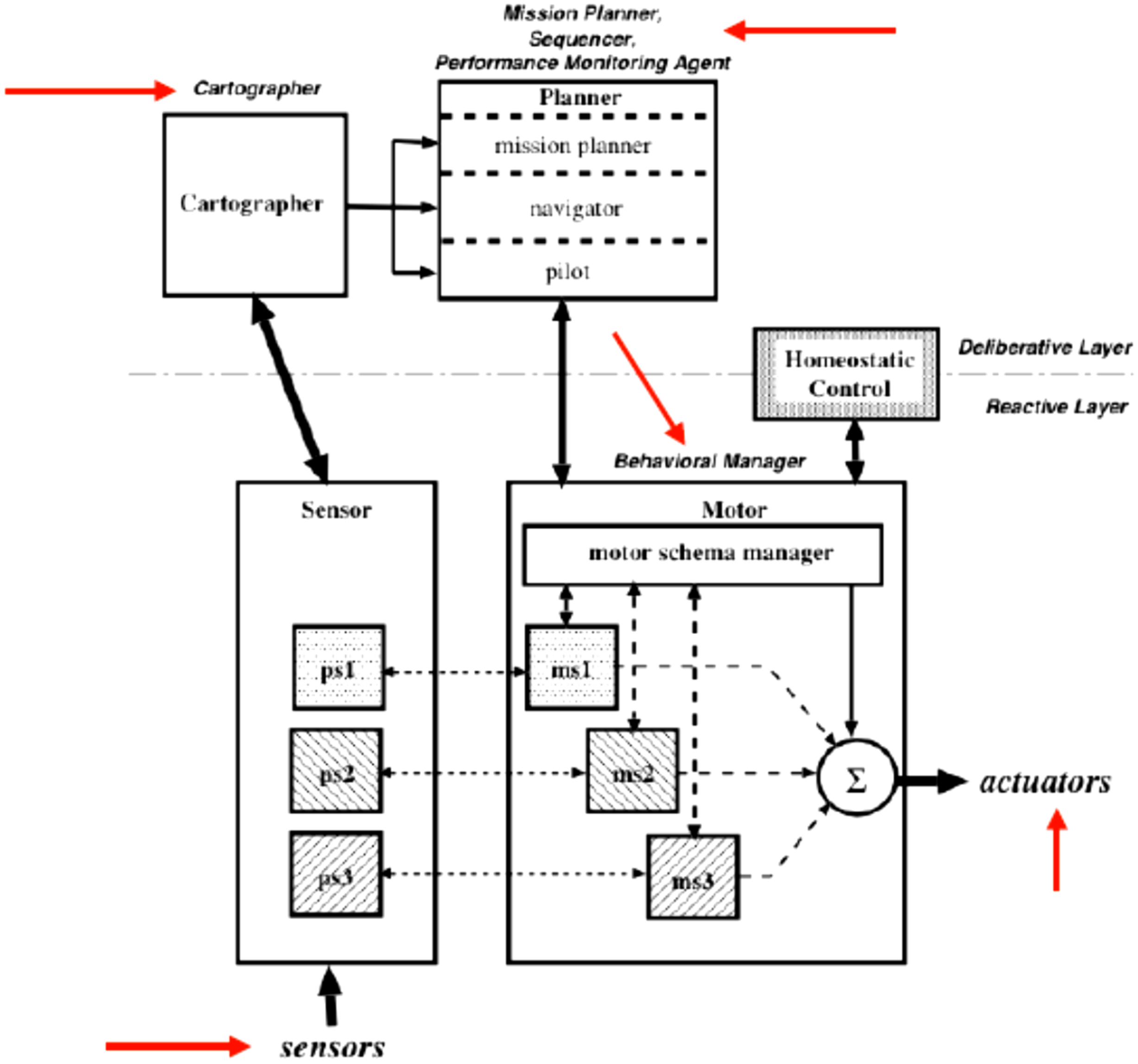
- Deliberative:
  - Representations
  - Models
  - Planning
- Reactive
  - Real-time
  - Multiple goals
  - Robustness
  - Flexibility
  - Modularity



Example of hybrid architecture: *AURA*

Strategic planning / reasoning

Low(er)-level controls and behaviors



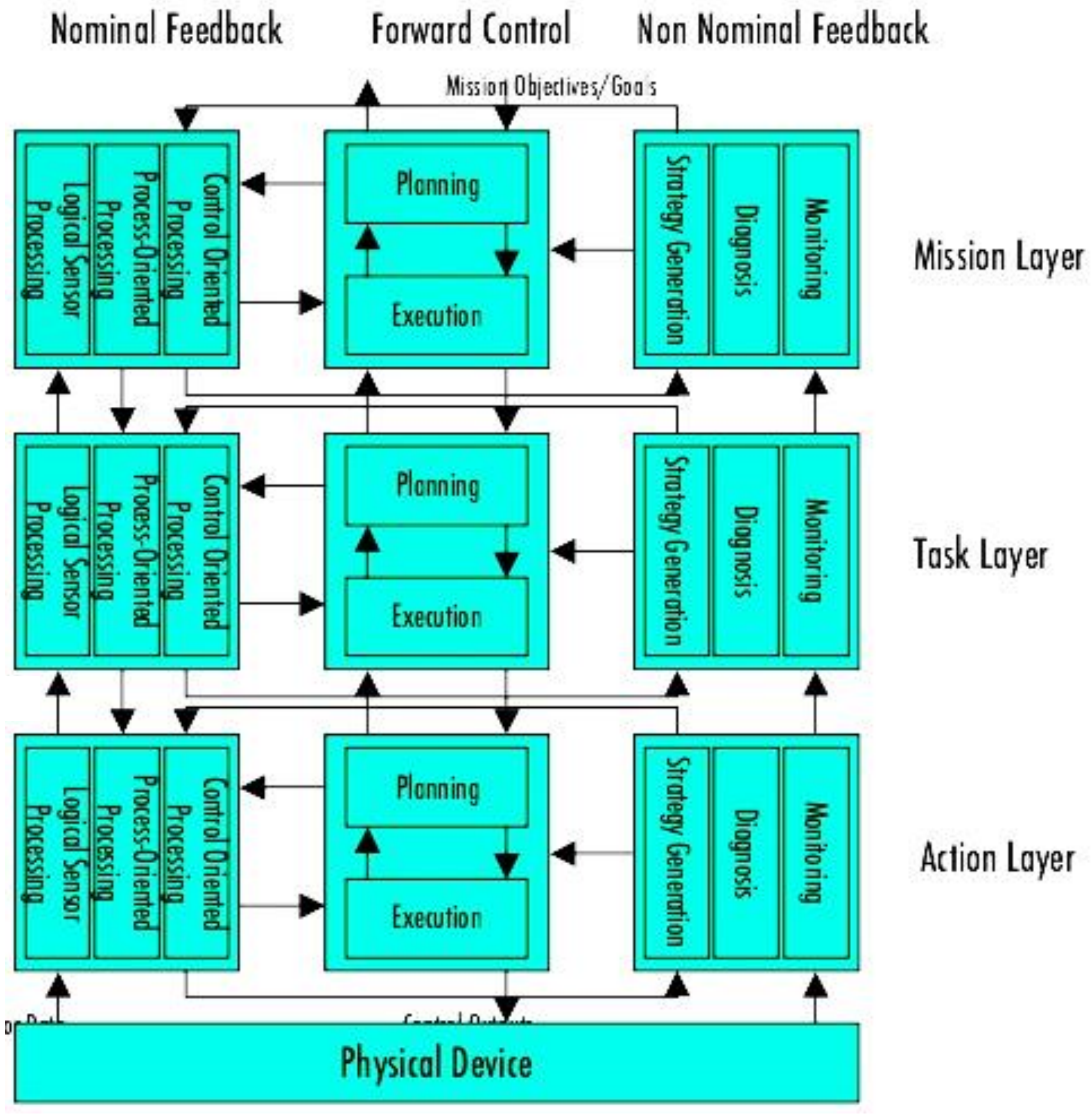
R. Arkin, T. Balch, *AuRA: principles and practice in review*, Journal of Experimental and Theoretical Artificial Intelligence 9 (2–3): 175–189, 1997

An example in space robotics

ESA has developed a hybrid architecture named Functional Reference Model (FRM).

It uses three layers:

- Mission layer to perform decisional planning for high lever objectives
- Task layer handling in reactive way the tasks using pre-defined activities / behaviors
- Action layer implements control schemes to achieve stability and the reflexes to adapt the system



Source: G. Visentin. Autonomy in ESA Planetary Robotics Mission. 2007

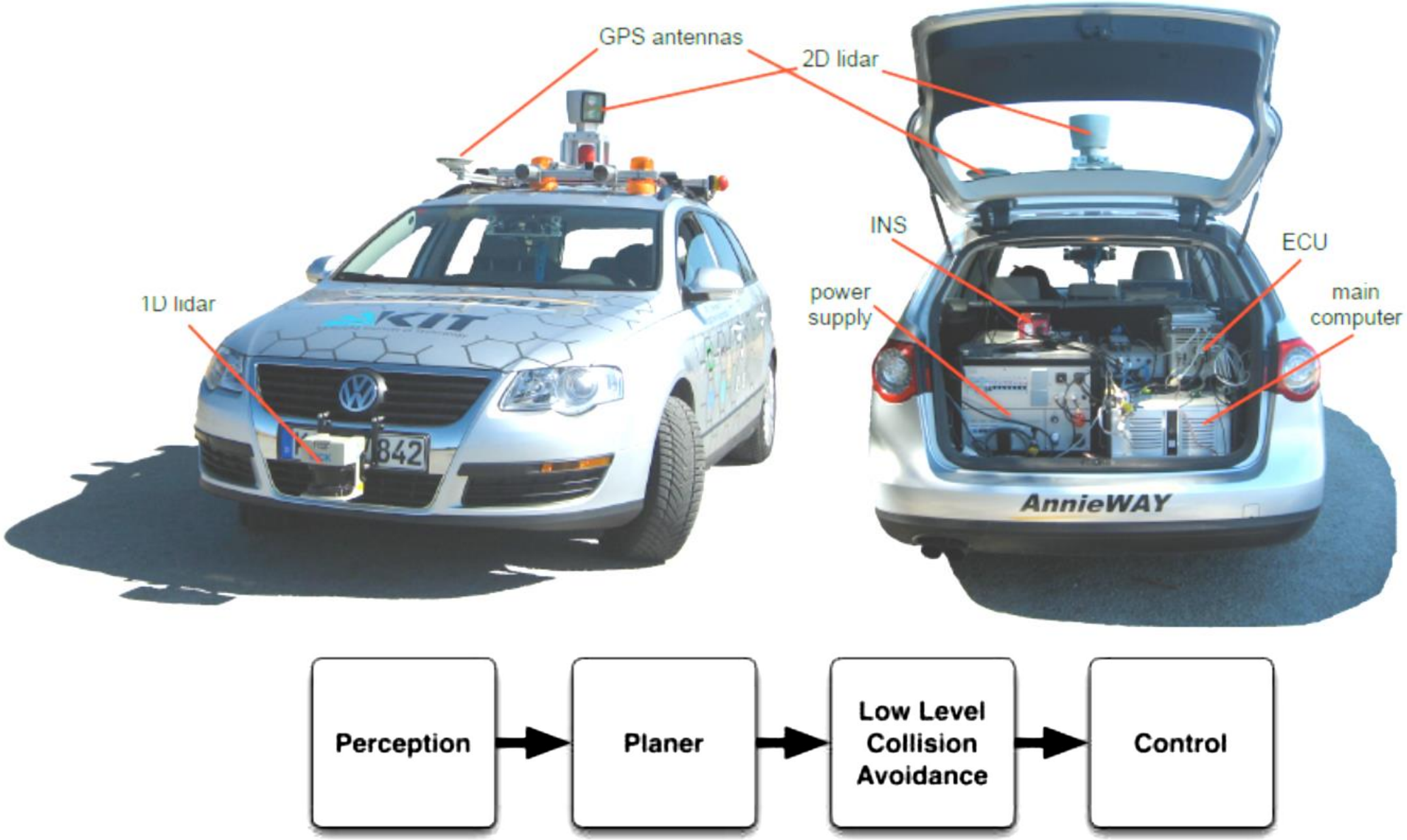
Examples from the DARBA Urban Challenge 2007

Annieway
Karlsruhe/Munich
not finished

Boss
Carnegie Mellon
1st place

Junior
Stanford
2nd place

Team Annieway (Karlsruhe/Munich)

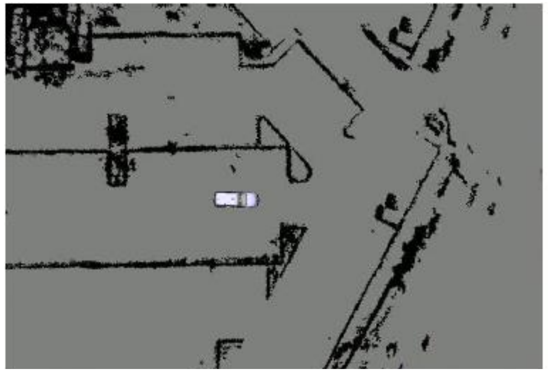


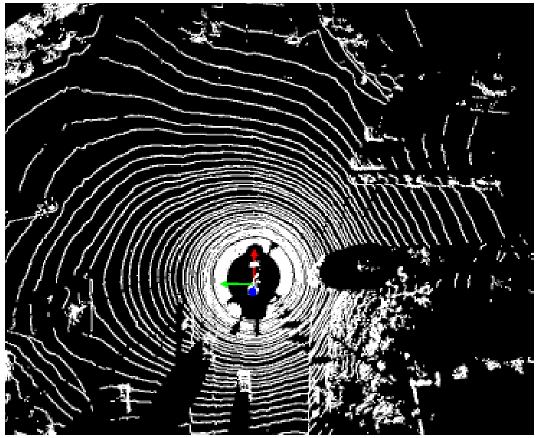
T. Gindele and D. Jagszent, "Design of the planner of Team AnnieWAY's autonomous vehicle used in the DARPA Urban Challenge 2007," *Intelligent Vehicle Symposium*, 2008

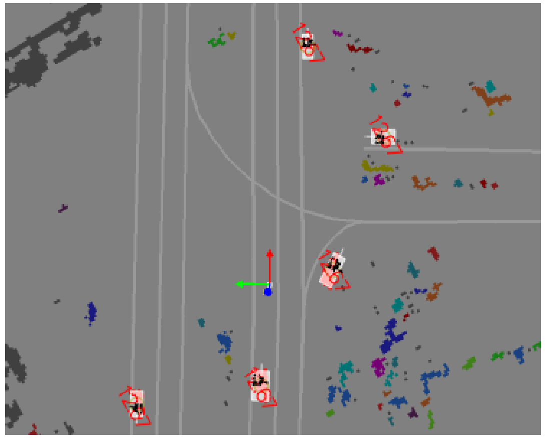
Team Annieway (Karlsruhe/Munich)

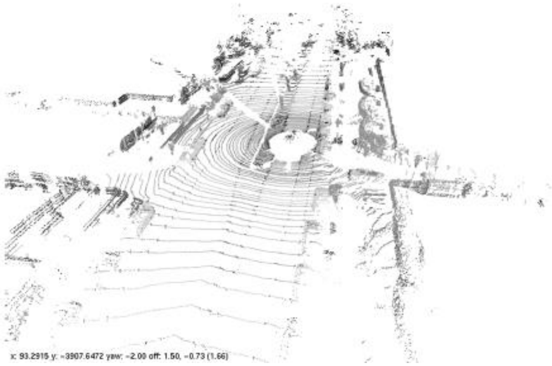
Perception performs several tasks simultaneously

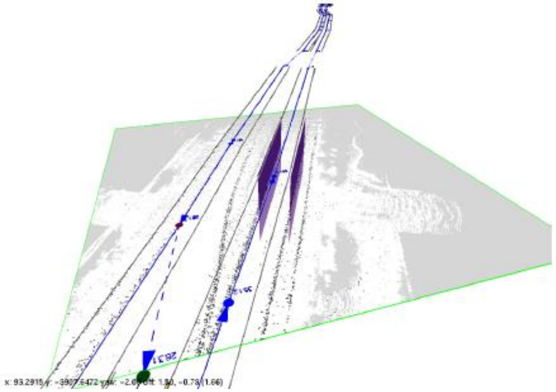
- Environment mapping through 3D lidar (Occupancy Grid mapping)
- Tracking of dynamic objects (Occupancy grid and Kalman Filter)
- Line marker detection (Combined lidar range and intensity)







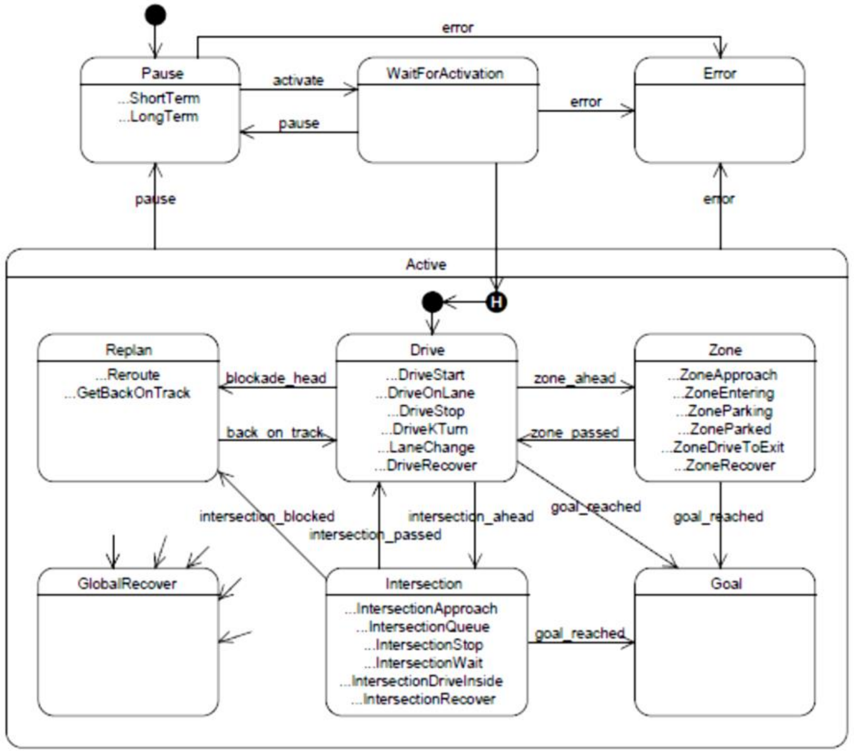




Team Annieway (Karlsruhe/Munich)

High-level state machine with several states

- Regular driving on lanes
- Turning at intersections with oncoming traffic
- Lane changing maneuvers
- Vehicle following and passing
- Following order of precedence at 4-way stops
- Merging into moving traffic



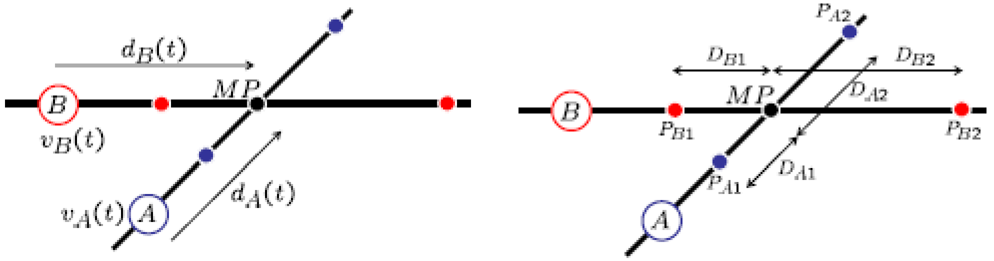
Mission planning by A* on roadgraph

T. Gindele and D. Jagszent, "Design of the planner of Team AnnieWAY's autonomous vehicle used in the DARPA Urban Challenge 2007," *Intelligent Vehicle Symposium*, 2008

Team Annieway (Karlsruhe/Munich)

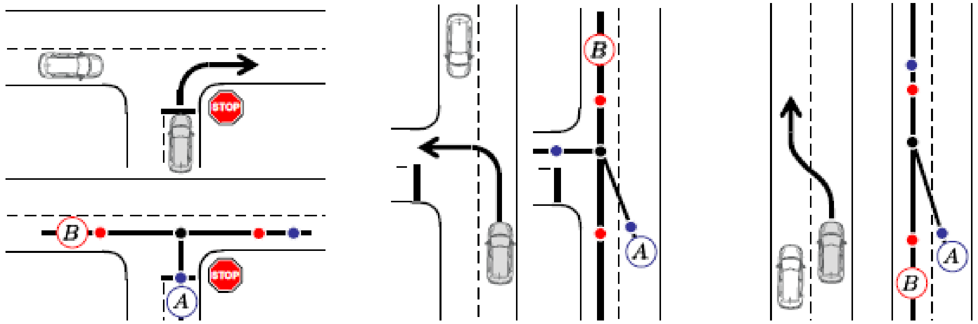
Situational awareness module enhances capabilities of the state-machine

- Enforce spatial and temporal gap to moving objects along lanes
- Simple feasibility check of maneuver



Assumptions:

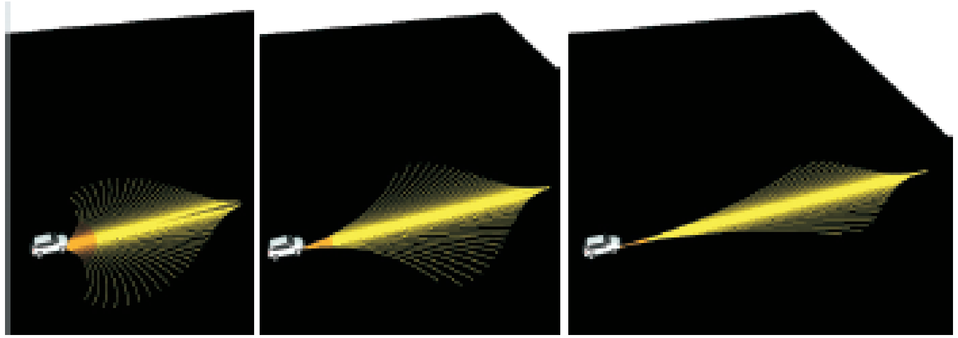
- Constant velocity of other traffic participants
- Constant acceleration of ego vehicle until desired velocity



Team Annieway (Karlsruhe/Munich)

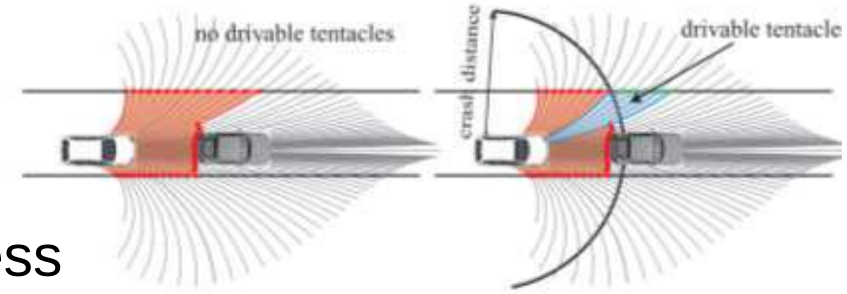
Pre-computed sets of motion primitives for different initial velocities

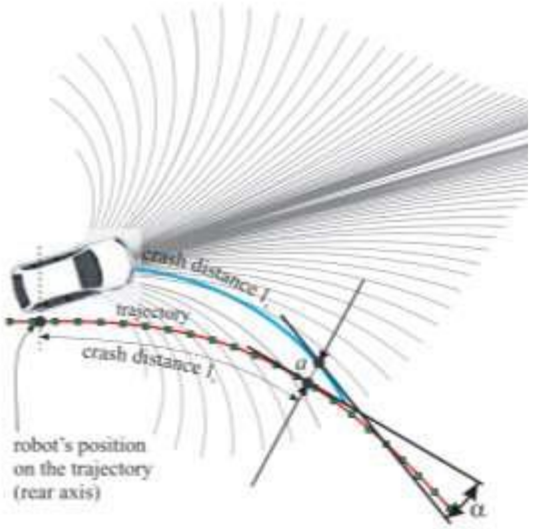
- Constant steering angles circular arcs (Dynamic Window Approach)
- Arc lengths shorter for high curvatures to avoid endpoints behind vehicle



Cost Function:

- Clearance: distance to closest obstacle along trajectory
- Flatness: averaged terrain flatness over support area
- Trajectory: alignment of trajectory with a reference path



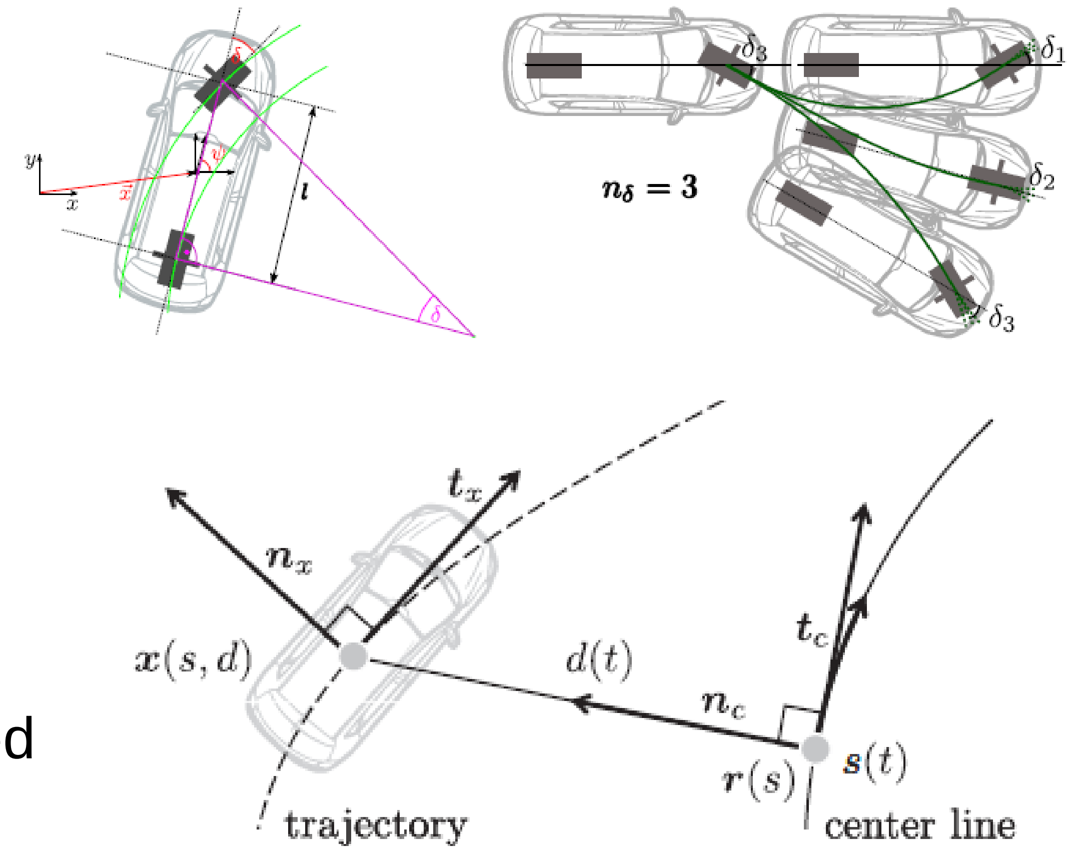


Team Annieway (Karlsruhe/Munich)

Trajectory planning is performed on a search graph

- A* search algorithm
- Single track motion primitives
- Two heuristic combined
  - Kinematic constraints (close)
  - Voronoi diagram (far)

Plannig assume two independent integrators for the longitudinal and lateral control and generates two sets of trajectories then merged



M. Werling, et al., "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds", IJRR, Dec. 2011.

Team Annieway (Karlsruhe/Munich)

Team AnnieWay

2007

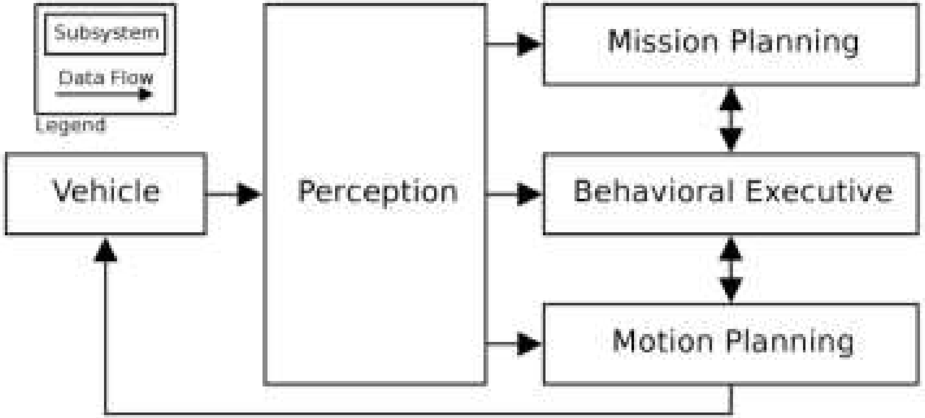
Visualization of AnnieWAY's Autonomous Run
in the DARPA Urban Challenge Final

http://his.anthropomatik.kit.edu

Team Boss (Carnegie Mellon)

Team Boss uses and hybrid architecture too

- Mission Planning
  - In charge of computing expected time to reach the waypoints
- Behavioral Executive
  - High-level management (follow lane, park)
  - Goal-assignment
  - On-road driving
  - Lane-change maneuvers
  - Intersection handling
- Motion Planning
  - On-road driving
  - Unstructured zone navigation



C. Baker and J. Dolan, “Traffic interaction in the urban challenge: Putting boss on its best behavior”, IEEE Int. Conf. Intell. Robot. Syst., 2008.

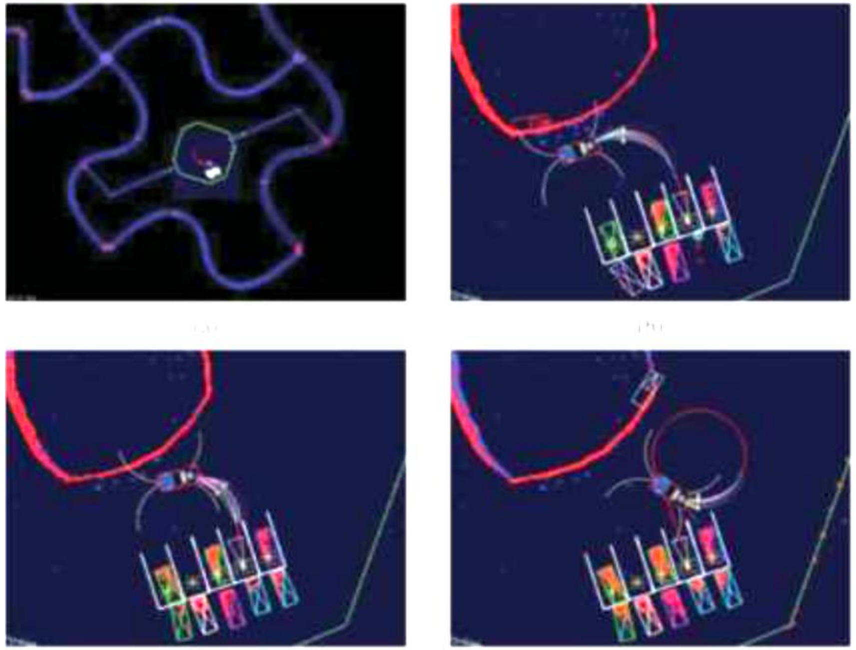
Team Boss (Carnegie Mellon)

Motion Planning in unstructured areas

- Anytime D* graph-search
- Multires 4D state-lattice (x, y , θ, v)
- Maximum-of-two heuristic
- Set of concatenations of two motion primitives (diverging / returning to path) for control

Motion Planning on road

- Take the lane center
- Motion primitives with final lateral offset to reference path



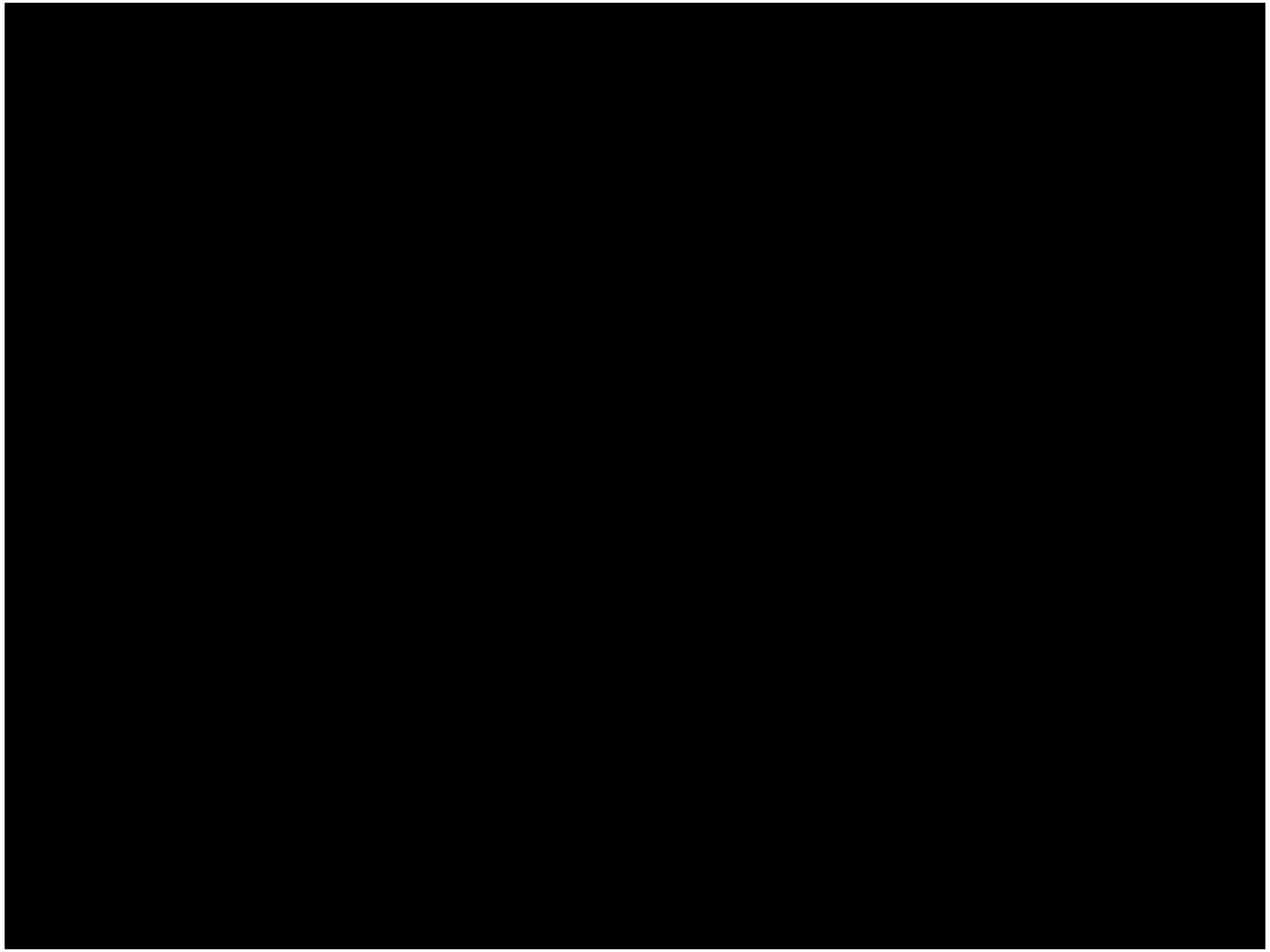
D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments", Journal of Field Robotics, vol. 25, no. 11–12, 939–960, 2008

Team Boss (Carnegie Mellon)

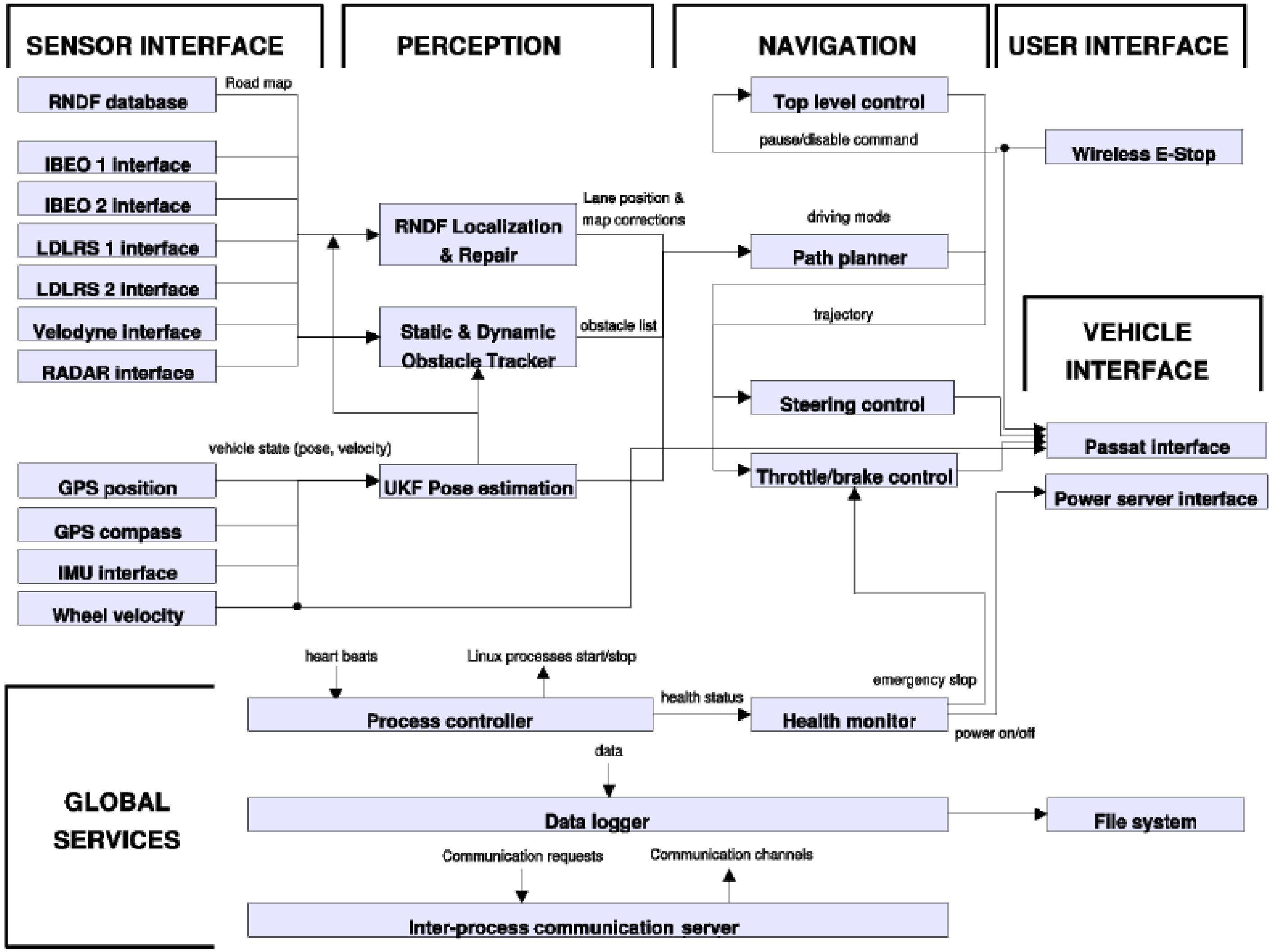
Success recipies:

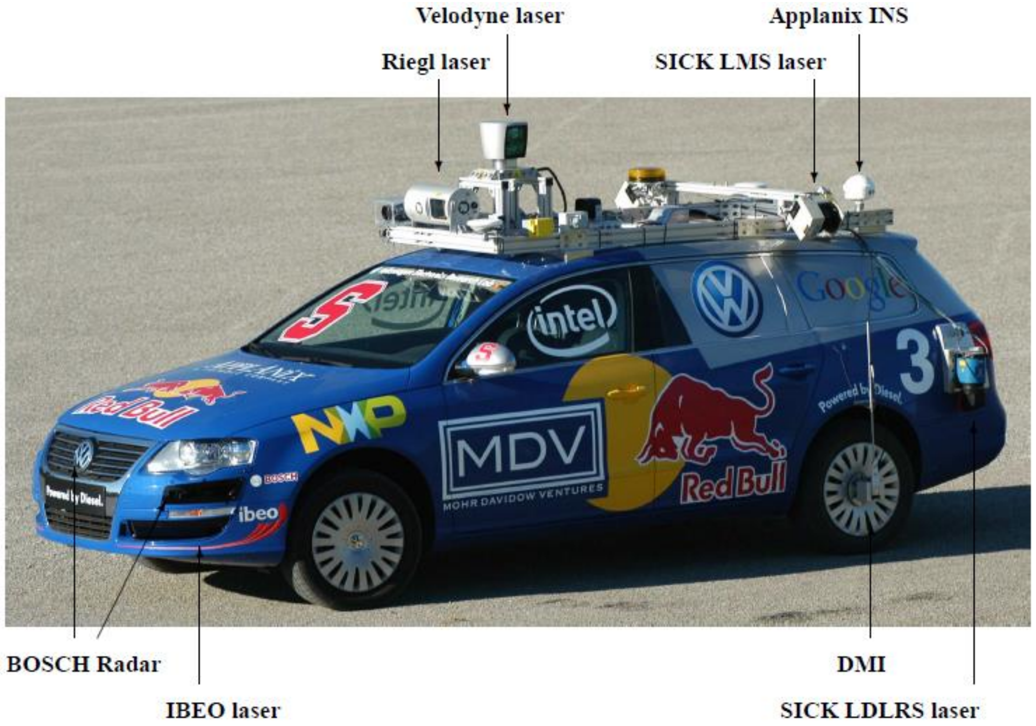
- Fast computation ensure smooth behavior
  - Preprocessing suggested wherever possible
- Detailed global planning stage increases system performance
  - Minimize divergence between planning stages
- Accurate vehicle modeling minimizes divergence between planning & execution
  - Higher speeds become safely driveable

Team Boss (Carnegie Mellon)



Team Junior (Stanford)

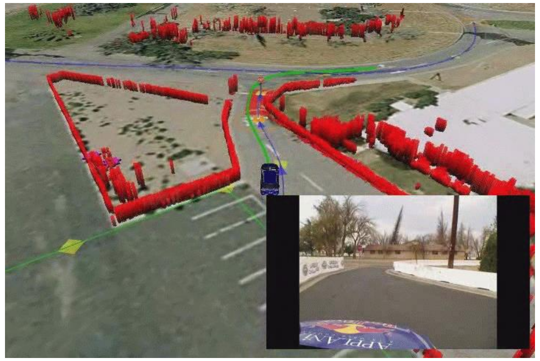


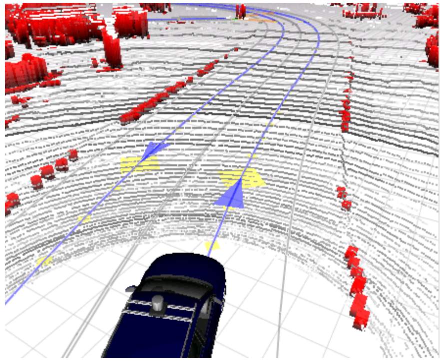


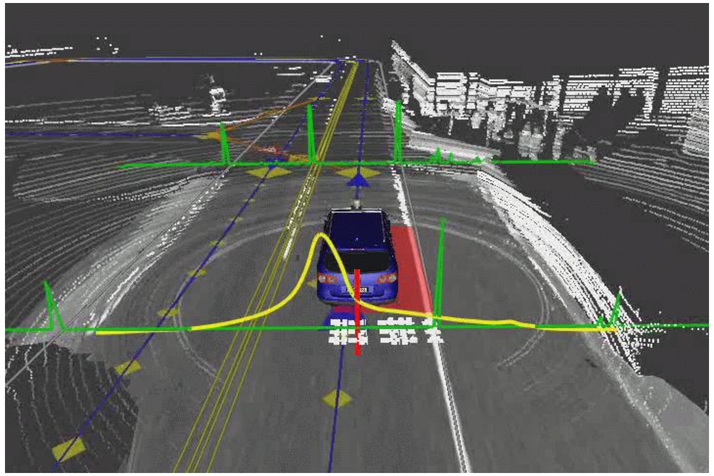
Team Junior (Stanford)

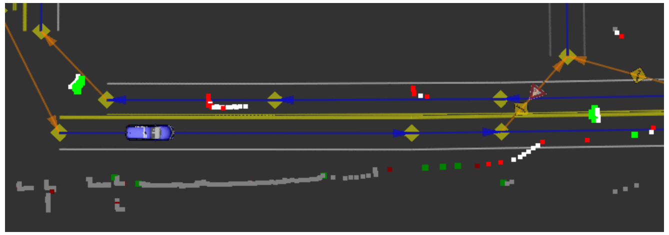
Perception performs several (usual) tasks simultaneously

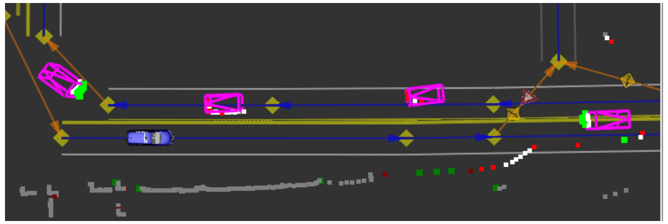
- Obstacle detection (Velodyne + IBEO)
- Grid mapping by evidence accumulation
- Object detection by scan differencing
- Localization on road network description file

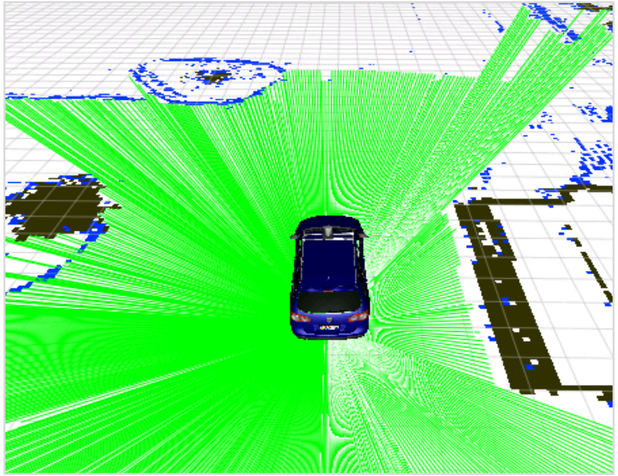


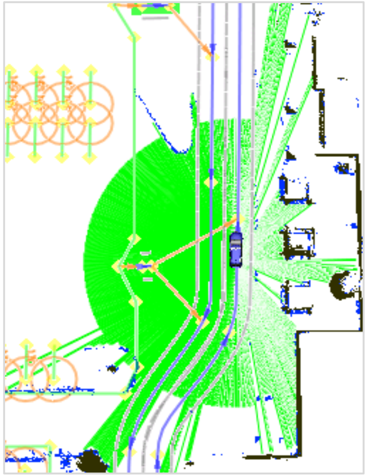








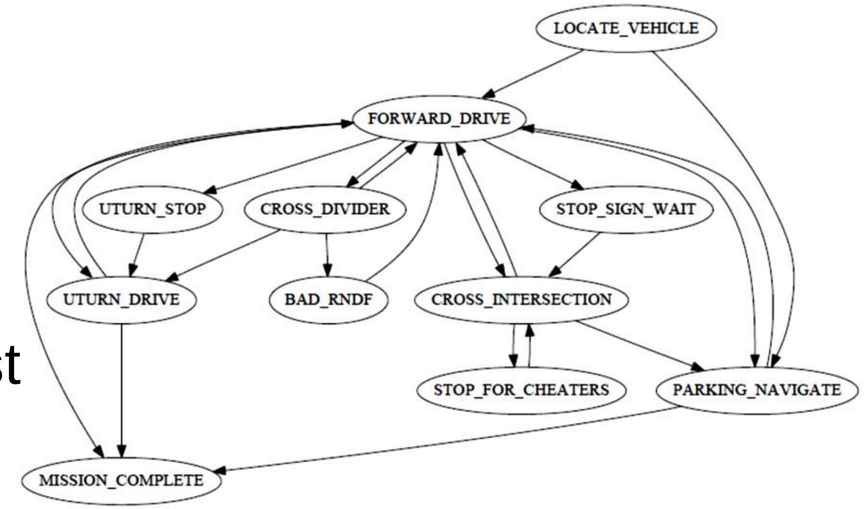


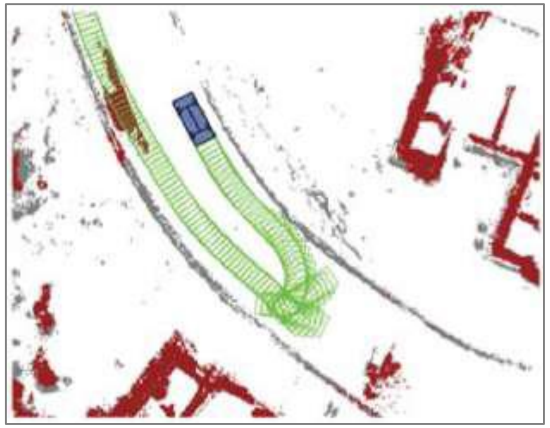
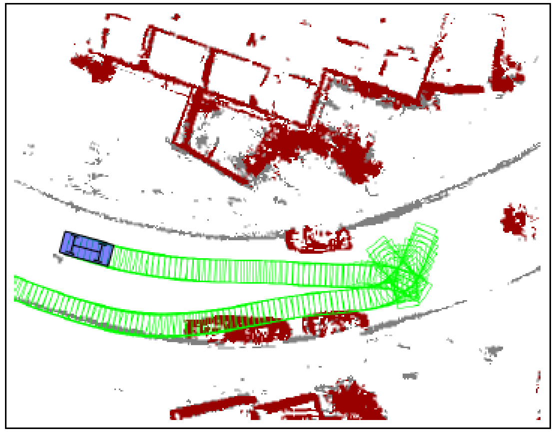
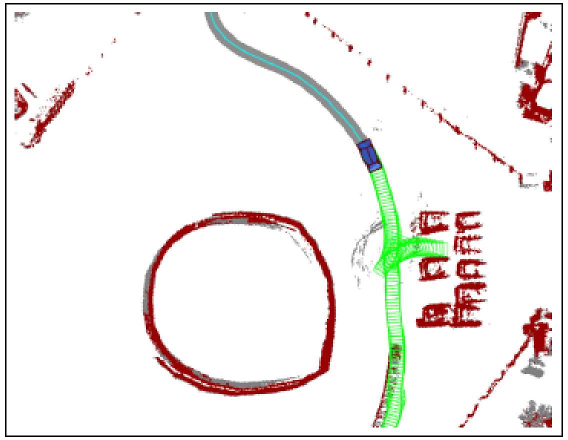


Team Junior (Stanford)

Motion planning

- Hybrid architecture based on a state machine
- Hybrid A* for navigation in unstructured space using maximum-of-two heuristic
- Graph-search on roadmap provides location cost
- Post-smoothing of paths by conjugate gradient



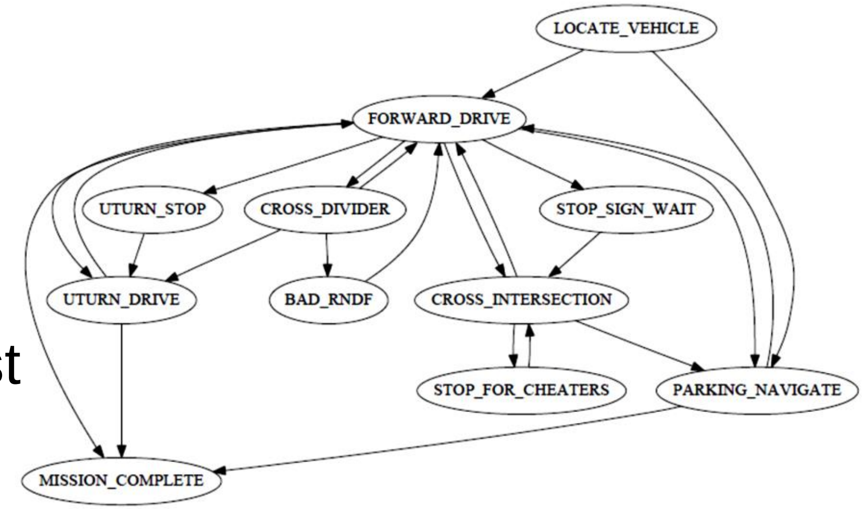





M. Montemerlo et al., "Junior: The Stanford entry in the Urban Challenge", Journal of Field Robotics, vol. 25, no. 9, pp. 569–597, 2008.

Team Junior (Stanford)

Motion planning

- Hybrid architecture based on a state machine
- Hybrid A* for navigation in unstructured space using maximum-of-two heuristic
- Graph-search on roadmap provides location cost
- Post-smoothing of paths by conjugate gradient

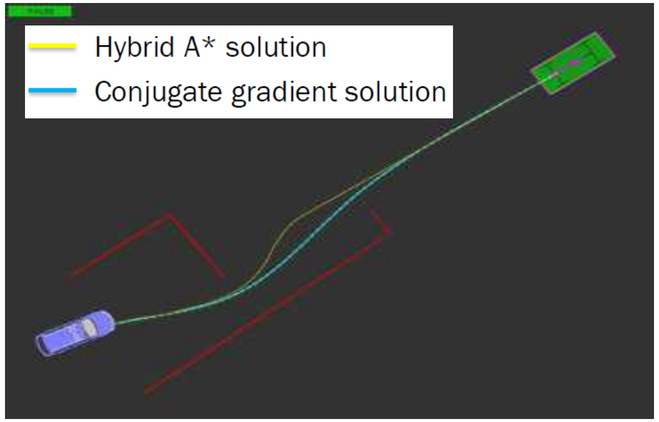


$$w_o \sum_{i=1}^{N} \sigma_o \left(|\mathbf{x}_i - \mathbf{o}_i| - d_{\max}\right) + w_\kappa \sum_{i=1}^{N-1} \sigma_\kappa \left(\frac{\Delta\phi_i}{|\Delta x_i|} - \kappa_{\max}\right) + \; + w_s \sum_{i=1}^{N-1} (\Delta \mathbf{x}_{i+1} - \Delta \mathbf{x}_i)^2$$

Obstacle distance penalty

Maximum curvature violation penalty

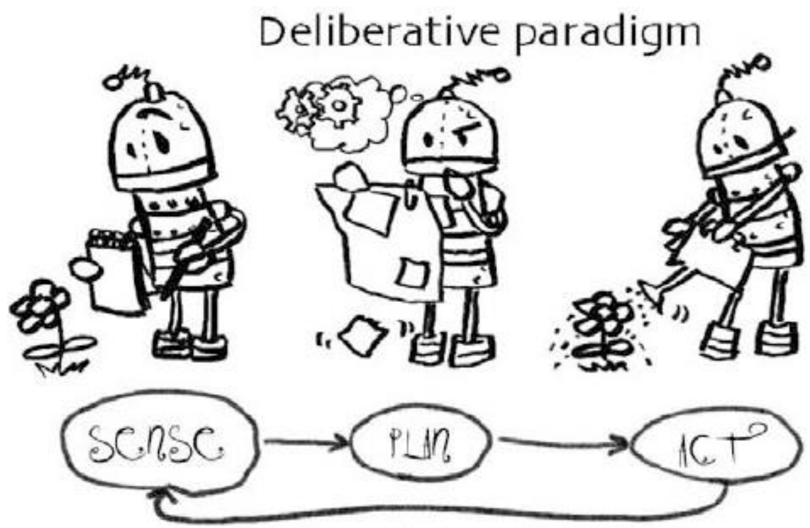
Maximum acceleration violation penalty

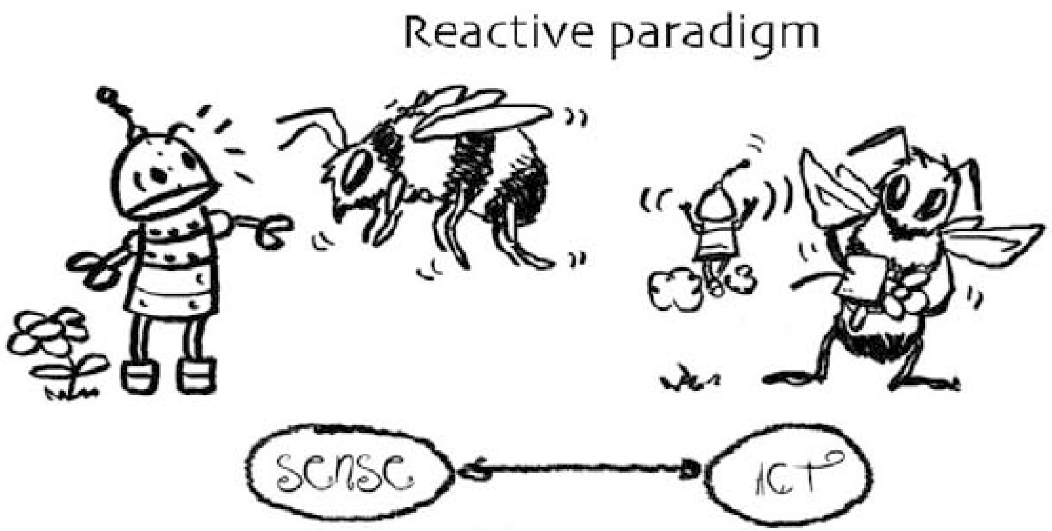


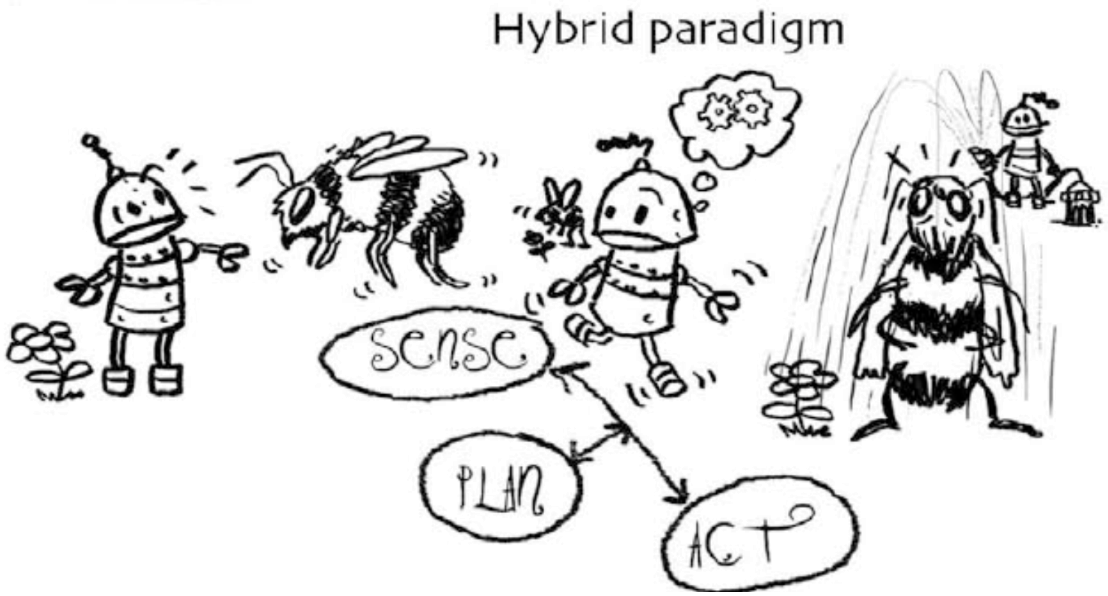
M. Montemerlo et al., "Junior: The Stanford entry in the Urban Challenge", Journal of Field Robotics, vol. 25, no. 9, pp. 569–597, 2008.

Team Junior (Stanford)

Let’s wrap up!







More to come if you include learning …

Wrap-up slide on "Robots Cognitive Architectures"

What should remain from this lecture?

- What a control architecture is for and why it is useful
- Difference between Deliberative / Reactive / Hybrid approaches
- What the «Sense / Plan / Act» paradigm is

References

- R. Arkin, Behavior-Based Robotics, MIT Press, 1998
- R. Murphy, An Introduction to AI Robotics, MIT Press, 2000
- M. Mataric, The Robotics Primer, MIT Press, 2007
- J. Jones, A Practical Guide to Behavior-based Robotics, McGraw-Hill, 2004