Agent Architectures

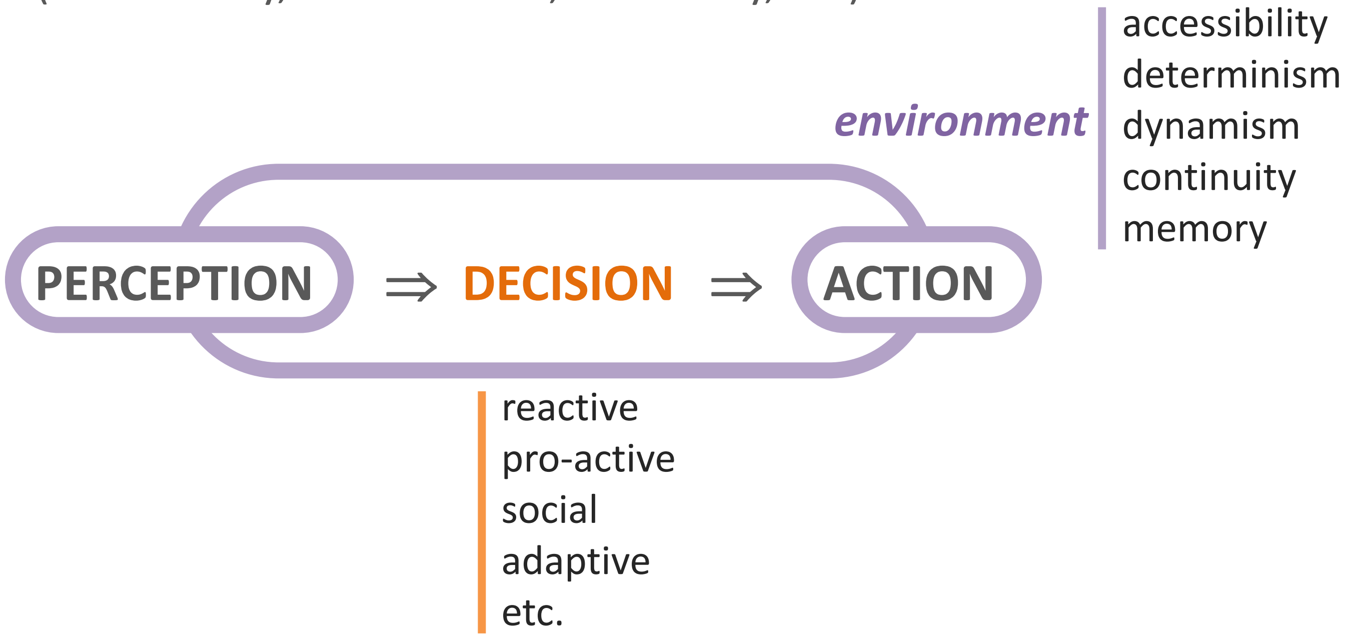
Outline

- **Introduction to agent architectures**
- Abstract architectures for agents
- Deductive reasoning agents
- Agents as intentional systems
- Reactive agents
- Hybrid architectures

Recalling concepts

AGENT & ENVIRONMENT

- **Sense – Decide - Act**
- **Agent properties** (autonomous, reactive, pro-active, social, adaptive, etc.)
- **Social Ability** (cooperation, coordination, negotiation)
- **Environment properties** (accessibility, determinism, continuity, etc)



The architectural stance

Modular thinking in Computer Science

- <u>System components</u>
  - perception
  - decision making
    - planning
    - learning
  - action

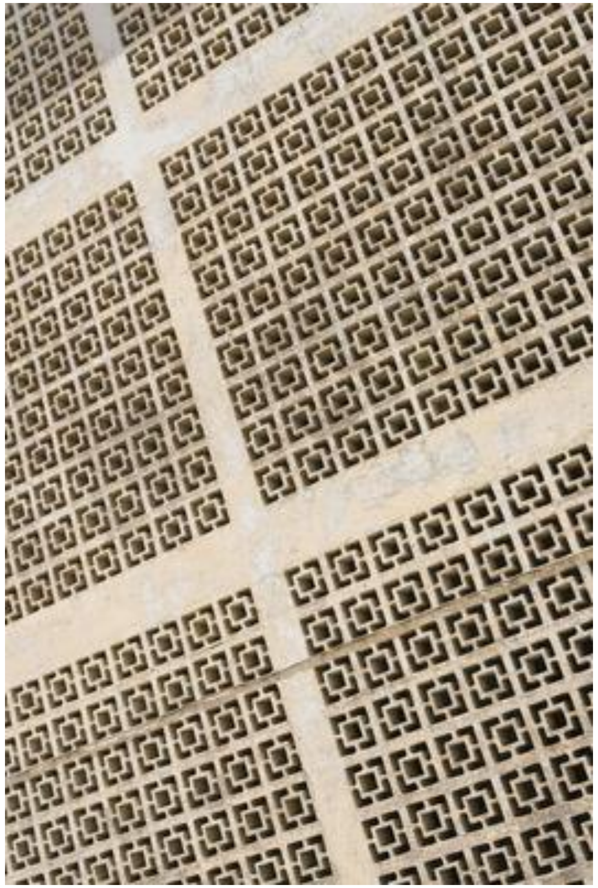
⇒ *an architecture, please!*

- <u>Component interactions</u>
  - between internal components
  - between agent and environment
  - between agents

Agent architectures

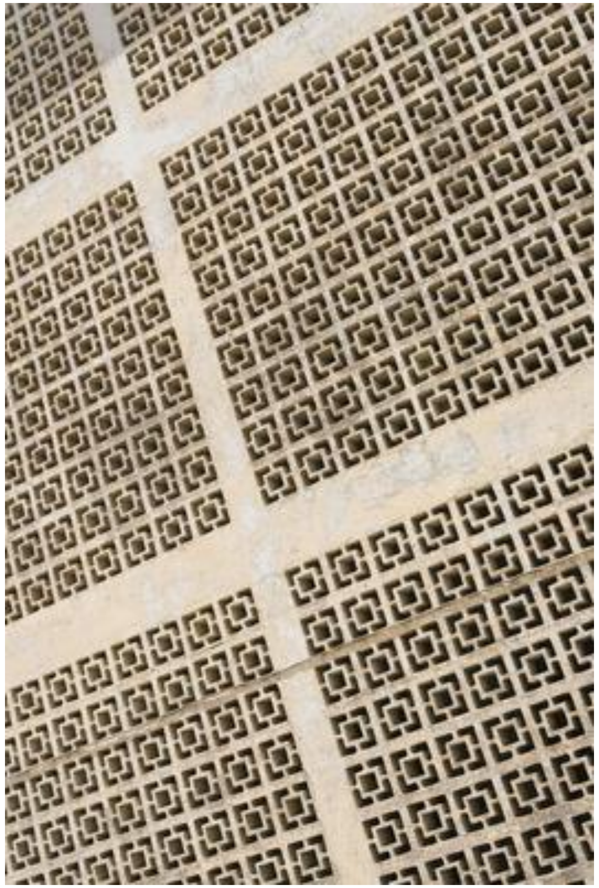
- **What is an agent architecture?**
  - Principles describing agent behavior with
    - formal/abstract view of agents
    - map of the internals (control-flow)

- **What is the goal of an architecture?**
  - Guide agent design and engineering



Agent architectures

- **What is an abstract architecture?**
  - Common principles describing agent behavior independently of their specificities
  - A template for the construction of agents



Outline

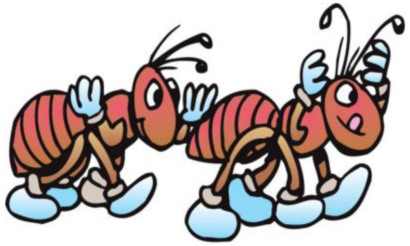
- Introduction to agent architectures
- **Abstract architectures for agents**
- Deductive reasoning agents
- Agents as intentional systems
- Reactive agents
- Hybrid architectures

Abstract architectures for agents

Let us make formal the abstract view of agents:

- ***Environment states***: (*finite*) set of (*discrete*) *states*

  $E = \{e_0, e_1, \ldots\}$



- *Agents have a set of **actions** (which transforms the environment's state) :*

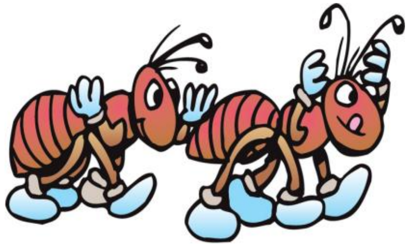
  $Ac = \{\alpha_0, \alpha_1, \ldots\}$

- ***Run**:* finite sequence of interleaved *states* and *actions*

  $r: e_0 \xrightarrow{\alpha_0} e_1 \xrightarrow{\alpha_1} e_2 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{n-1}} e_n$

Abstract architectures for agents

Let:



- R be the set of **all such possible runs** (over E and Ac)
  where r, r' are members of R
- R^{Ac} is the subset of these that *end with an action*
- R^{E} is the subset of these that *end with an environment state*

Abstract architectures for agents

- A **state transformer** (environment changes):

$$\tau: R^{Ac} \rightarrow \wp(E)$$

Maps a run (ending in an action) to a set of possible environment states

Important points about this definition:

- History dependent
- Non-determinism

if $\tau(r)=\varnothing$, there are no possible successor states to r (system has *ended* its run)

Abstract architectures for agents

- An **environment** can now be fully defined as a triple

  $Env = < E, e_0, \tau >$ where:

  – E is a set of environment states

  – $e_0 \in E$ is the initial state

  – τ is a state transformer function

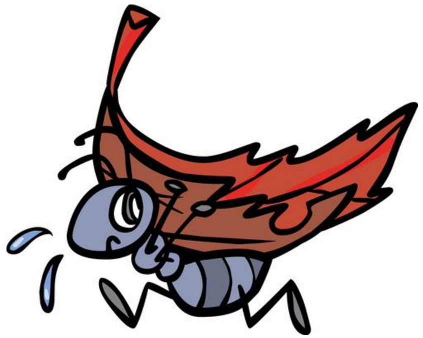
Abstract architectures for agents

- **Agent** is a function which maps runs to actions:

$$Ag: R^E \rightarrow Ac$$

  An **agent makes a decision** (i.e., action to perform) based on the **history** of system that it has witnessed to date (i.e., R^E)

- Let $\boldsymbol{AG}$ be the **set of all agents**, $Ag \in AG$



Abstract architectures for agents

- A **system** is a **pair with:**
  - **one (or more) agent(s)** and
  - **an environment**

- Any **system** has **a set of possible runs**

Abstract architectures for agents

- We denote the **set of runs of agent** $\boldsymbol{Ag}$ in **Environment** $\boldsymbol{Env}$ by

$$R(Ag, Env)$$

- We assume $R(Ag, Env)$ contains only **runs that have ended**

Outline

- Introduction to agent architectures
- Abstract architectures for agents
- **Deductive reasoning agents**
- Agents as intentional systems
- Reactive agents
- Hybrid architectures

The oldest agent architecture

- **1956–1985**: pretty much all agents designed within AI were ***symbolic reasoning*** agents (mostly *deductive reasoning*)
  - agents with *explicit logical reasoning* to decide what to do

- **1985–present**: problems with symbolic reasoning led to the so-called ***reactive** agents* movement (1985–present)

- **1990-present**: diverse alternatives, including ***hybrid*** architectures, combining the best of *deliberative* reasoning and *reactiveness*

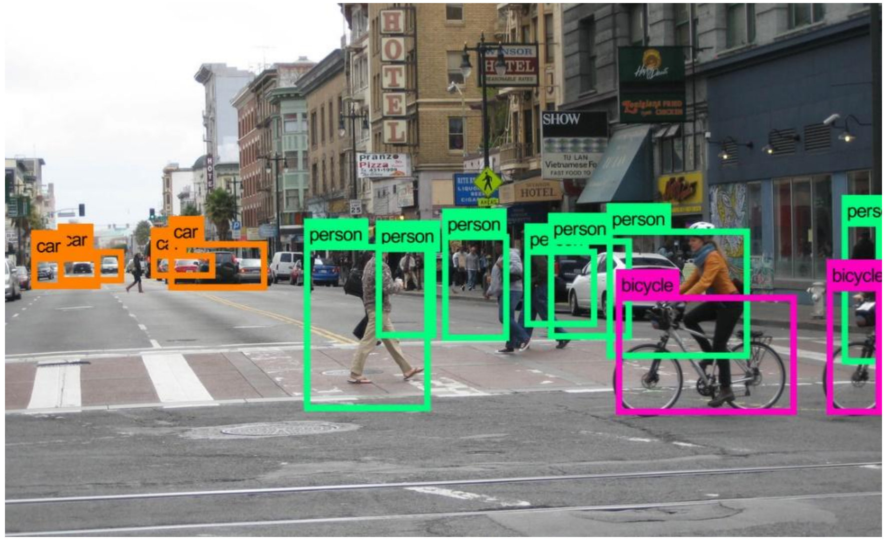
Deductive reasoning agents

- Classical approach for creating deductive agents:
  - **Agent as a knowledge-based system**
- This paradigm is known as **symbolic AI**

Deductive reasoning agents

Two key problems to be solved:

1. ***Transduction problem***:
   - translating real-world environment into an accurate, adequate symbolic description.



Fields: computer vision, speech understanding, learning…

Deductive reasoning agents

Two key problems to be solved:

2. ***Representation/reasoning problem***:
   - symbolically representing information
   - how to get agents to reason with this information

Fields: knowledge representation, automated reasoning, planning

Deductive reasoning agents

- ***Deductive reasoning agent*** (architecture) is one that:
  - contains an explicit **symbolic representation of the world**
  - **internal state** given by formulae (**predicate logic**)

- Example with first-order predicate logic:

```
isopen(valve221)
temperature(reactor4726, 321)
pressure(tank776, 28)
```

Deductive reasoning agents

- ***Deductive reasoning agent*** (architecture):
  - **analogous to beliefs in humans**:
    - internal state may include incorrect/outdated info
  - **makes decisions via *symbolic reasoning***
    - proving theorems without breaking axioms on what is possible

Agents with state

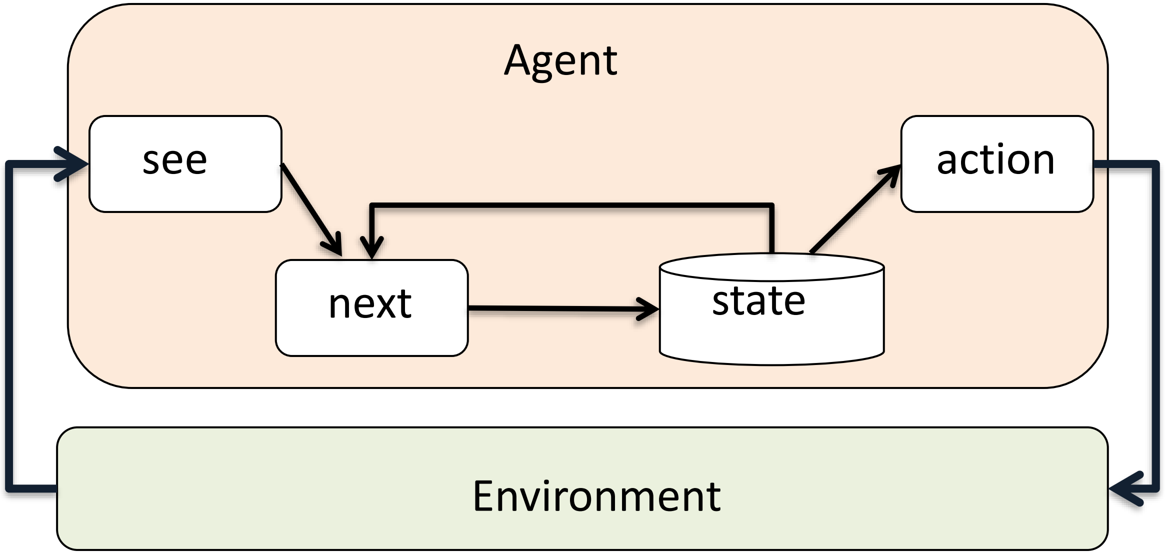
Agent decision:

- D (internal state = set of formulae or database)

– $see: S \to Per$ (observe the environment)

– $next: D \times Per \to D$ (update internal state)

– $action: D \to Ac$ (decision making with deduction rules)



Deductive reasoning agents

How can an **agent decide what to do using theorem proving**?

- Use <u>logic to encode a theory stating the *best* action to perform</u> in a given situation

Deductive reasoning agents

- Let:
  - ρ be this theory (typically **deduction rules**)
  - *DB* be the logical data (database) describing **current state** of the world
  - Ac be the **set of actions** the agent can perform
  - $DB \vdash_{\rho} \phi$ means that **formula ϕ can be proved from database *DB* using deduction rules ρ**

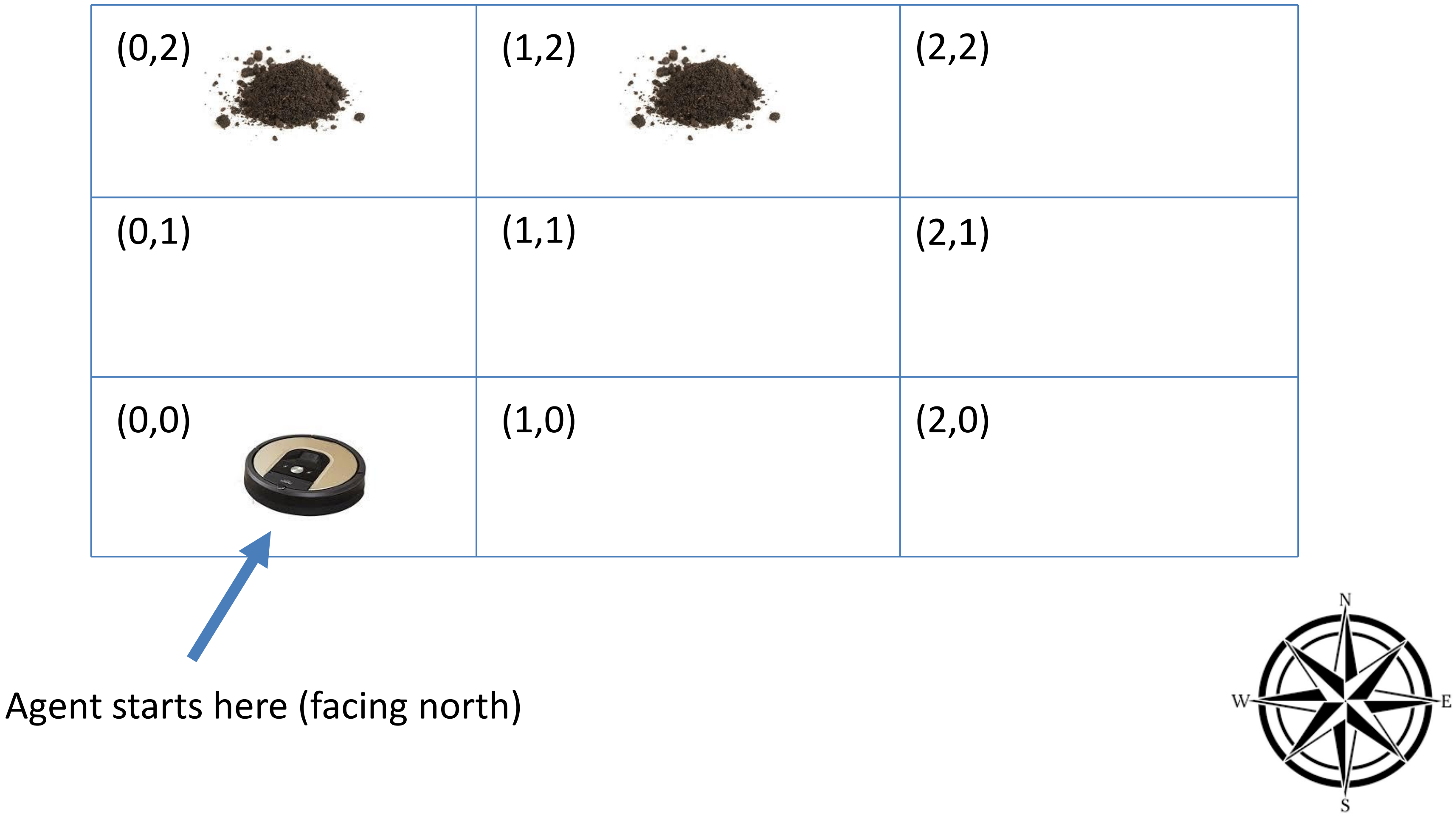
Deductive reasoning agents

- Agent's action selection function (i.e., $action: D \rightarrow Ac$) is defined in terms of its deduction rules

function ***action(DB:D)*** returns an action *Ac*
begin
/* *for each action, attempts to prove Do(a) from its database using deduction rules* */
for **each** $\boldsymbol{a} \in \boldsymbol{Ac}$ do
if *DB* $\vdash_{\rho} Do(a)$ then return a
end for
/* *attempts to find an action such that ¬Do(a) cannot be derived (i.e., not explicitly forbidden)* */
for **each** $\boldsymbol{a} \in \boldsymbol{Ac}$ do
if *DB* $\vdash_{\rho} \neg Do(a)$ = *false* then return a
end for
return *null* /* *no action found* */
end function

Vacuum world

Vacuum world



Vacuum world

- **Environment state**
  - S ={(0,0,$d_{0,0}$), (0,1,$d_{0,1}$), (0,2,$d_{0,2}$), (1,0,$d_{1,0}$), (1,1,$d_{1,1}$), (1,2,$d_{1,2}$), (2,0,$d_{2,0}$), (2,1,$d_{2,1}$), (2,2,$d_{2,2}$)}

- Agent can receive a **percept *dirt* or *null***
  - i.e., either **dirt under the agent** or not
  - Per= {*dirt*, *null*}

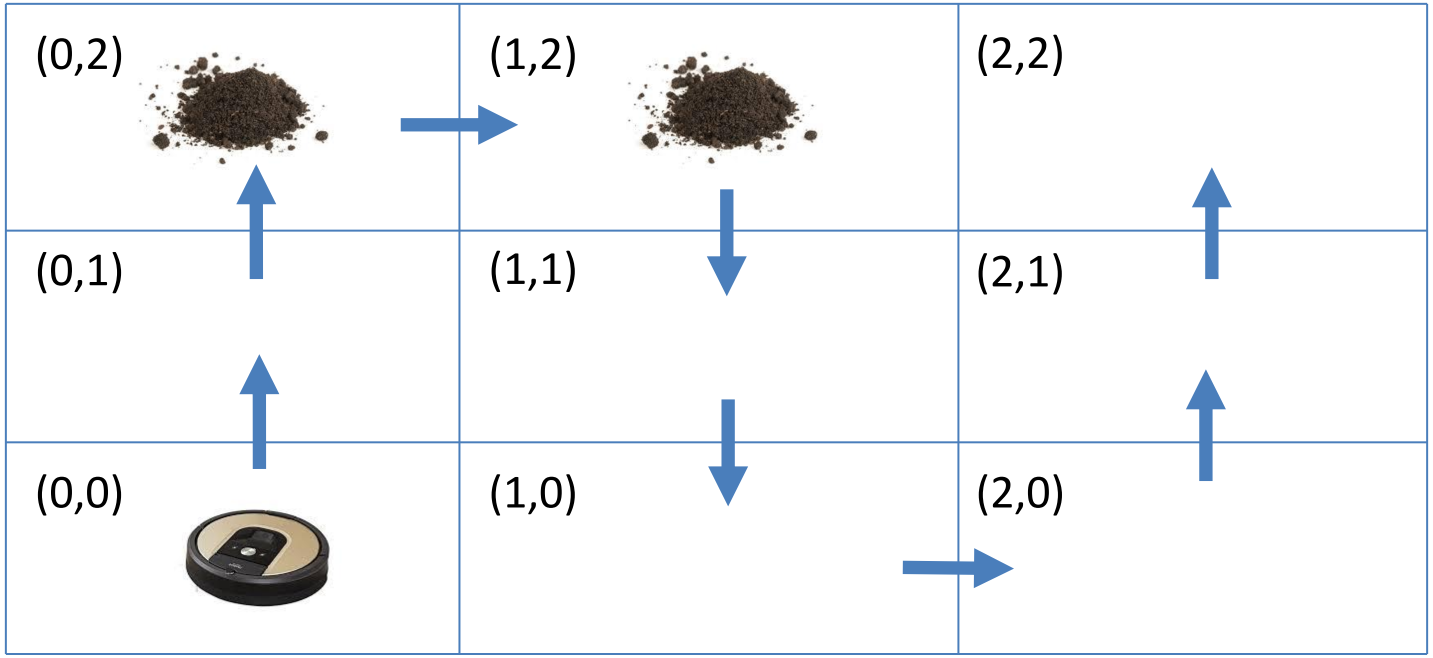
- **Actions**: *forward*, *suck*, or *turn* (right 90°)
  - Ac ={*forward*, *suck*, *turn*}

Vacuum world

- Internal state DB: three domain predicates
  - $In(x, y)$ – agent is at (x, y)
  - $Dirt(x, y)$ – there is dirt under the agent at (x, y)
  - $Facing(d)$ – the agent is facing direction d

Vacuum world

We need deduction rules for agent's behavior!



Vacuum world

- Deduction rules (agent's behavior):
  - $In(x, y) \wedge Dirt(x, y) \longrightarrow Do(suck)$
  - $In(0,0) \wedge Facing(north) \wedge \neg Dirt(0,0) \longrightarrow Do(forward)$
  - $In(0,0) \wedge \neg Facing(north) \wedge \neg Dirt(0,0) \longrightarrow Do(turn)$
  - $In(0,1) \wedge Facing(north) \wedge \neg Dirt(0,1) \longrightarrow Do(forward)$
  - $In(0,1) \wedge \neg Facing(north) \wedge \neg Dirt(0,1) \longrightarrow Do(turn)$
  - …

Final remarks: deductive reasoning agents

- **Agent's decision making strategy** – encoded as logical theory

- **Agent's action** – reduces to a problem of proof

- Logic-based approach are **elegant** and have **(clean) logical semantics**

Final remarks: deductive reasoning agents

- **Disadvantages:**
  - Inherent **computational complexity** of theorem proving
  - Cannot **operate effectively in time-constrained environments**
  - The **environment cannot change** while the agent is making a decision
  - Not easy to **represent and reason about complex and dynamic environments**

Outline

- Introduction to agent architectures
- Abstract architectures for agents
- Deductive reasoning agents
- **Agents as intentional systems**
- Reactive agents
- Hybrid architectures

Agents as Intentional Systems

Intentional stance

Develop **agent behaviors** in terms of ***mental states*** (beliefs, desires, wishes, hopes, …)

Agents as Intentional Systems

Examples:

- "Michael took his umbrella because he *believed* it was going to rain."
- "John worked hard because he *wanted* to obtain a PhD."

…such attitudes are called ***intentional notions***

Agents as Intentional Systems

This approach can be useful to model:

- **complex systems**
- systems whose **structure is incompletely known**

yes

Is it legitimate or useful to attribute beliefs, desires, and so on, to artificial agents?

How to implement agents using the Intentional Stance?

Intentional Systems are the base for deliberative agents,
which follow the *intentional stance* through practical reasoning

How to implement agents using the Intentional Stance?

- **Intentional systems** are the base for **deliberative agents**
  - This **agent architecture** has its origins in the **philosophical work** of Bratman:
    - Michael E. Bratman. *Intention, Plans and Practical Reason*. Harvard University Press, 1987.

How to implement agents using the Intentional Stance?

“Practical reasoning is a matter of ***weighing conflicting considerations*** **for and against competing options**, where the relevant considerations are provided by what the **agent *desires/values/cares* about and what the agent *believes***.” [Bratman, 1990]

Practical reasoning

Practical Reasoning = Deliberation + Means-Ends Reasoning

- ***Deliberation***: deciding *what* state of affairs an agent wants to achieve from (possibly conflicting) desires

- ***Means-Ends Reasoning***: deciding *how* an agent wants to achieve these states of affairs

The B.D.I. model

- Bratman's philosophical work was used as an **inspiration for creating an architecture for intelligent agents** based on the mental attitudes of:
  - ***beliefs***
  - ***desires***
  - ***intentions***

The B.D.I. model

- **Beliefs**
  - Information about the environment, other agents, and itself
- **Desires**
  - Desires/goals are state of affairs to achieve
- **Intentions**
  - Commitments to achieving particular goals

Intentions in practical reasoning

Intentions are stronger than mere desires:

"My desire to play basketball this afternoon is merely a potential influencer of my conduct.

It must be viewed with my other relevant desires…

Once I intend to play basketball this afternoon, the matter is settled: I normally need not continue to weigh the pros and cons.

When the afternoon arrives, I will normally just proceed with my intentions." (Bratman, 1990)

Intentions in practical reasoning

And **Intentions drive means-ends reasoning**

- They *lead to action* because I *attempt to achieve* them
- I try to decide *how* to achieve them
- If one course of action *fails*, I usually *attempt others*

Intentions in practical reasoning

For example:

- *You might consider a career as an academic or a career in industry (deliberation)*
- *You have to decide the career (deliberation)*
- *Your decision is to be an academic (intention/state of affairs)*
- *You make a plan: apply for PhD program, get a PhD, etc.*
  - *You decide how to achieve the state of affairs (means-end reasoning)*

Intentions in practical reasoning

Property: **Intentions persist**

- *I do not give up without good reason*

E.g., If your intention is to become an academic, then you should persist with this intention

Intentions in practical reasoning

Property: **Intentions constrain future deliberation**

- *I will not entertain options that are inconsistent*

filter of admissibility

E.g., If I have an intention to write a book, so I cannot consider the option of partying every night

Intentions in practical reasoning

Property: **Intentions influence beliefs**

- *Intentions are closely related to beliefs about the future*

E.g., If you intend to become an academic, then you should believe that, assuming some background conditions, you will indeed become an academic

Deliberation and belief revision

So how do we model ***deliberation in agents***?

- **Belief revision function**
  - Update beliefs with sensory input and previous belief

- **Function to generate options**
  - Use beliefs and existing intentions to generate options (=desires)

- **Filtering function**
  - Choose between competing alternatives and commit to their achievement

Deliberation and belief revision

So how do we model ***deliberation in agents***?

- revise the agent's beliefs (belief revision function):

$$brf: 2^{Bel} \times Per \rightarrow 2^{Bel}$$

- produce the agent's desires/options (option generation function):

$$options: 2^{Bel} \times 2^{Int} \rightarrow 2^{Des}$$

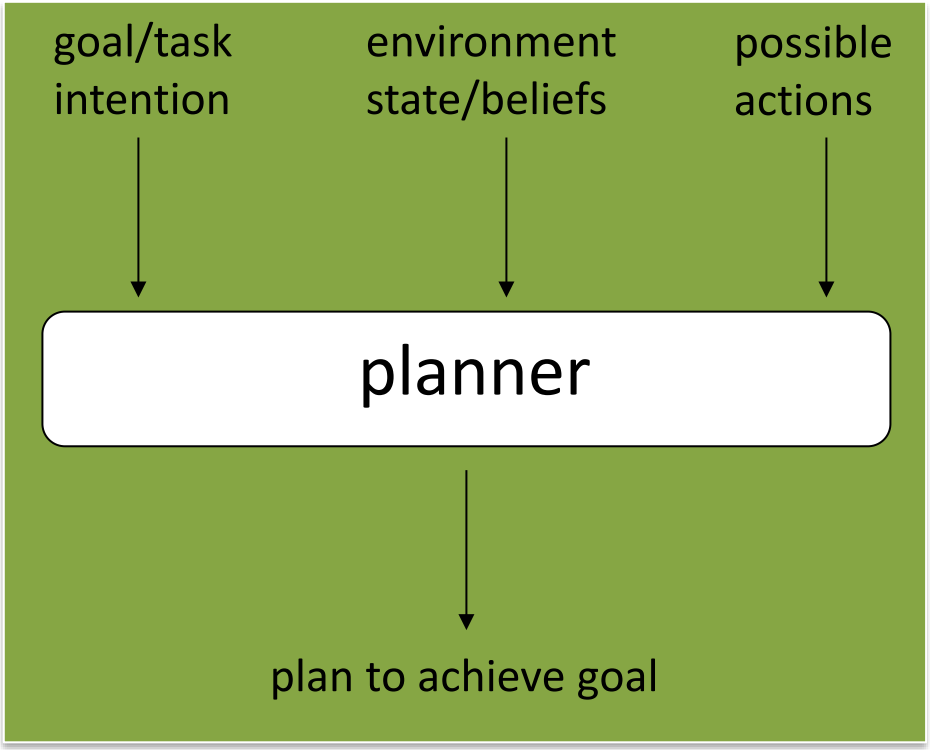
- select the *best* option(s) for the agent to commit to (filter function):

$$filter: 2^{Bel} \times 2^{Des} \times 2^{Int} \rightarrow 2^{Int}$$

Means-Ends reasoning

An agent's means-ends function

$$plan: 2^{Bel} \times 2^{Int} \times 2^{Ac} \rightarrow Plan$$



Implementing a practical reasoning agent

Decision-making is a *loop*:

1. *Observe* the world and *update beliefs*
2. *Deliberate* to decide the *intention(s)*
   - determine available *options*
   - *filter*
3. Use *means-ends reasoning* to find a *plan* for the intention(s)
4. *Execute* the plan
5. Return to 1

Commitments

How committed an agent should be to its intention?

How long should an intention persist?

- A commitment implies ***temporal persistence***.
- But to what extent?

When do I give up pursuing an Intention?

Commitment strategies

- Blind commitment
- Single-minded commitment
- Open-minded commitment

Blind commitment

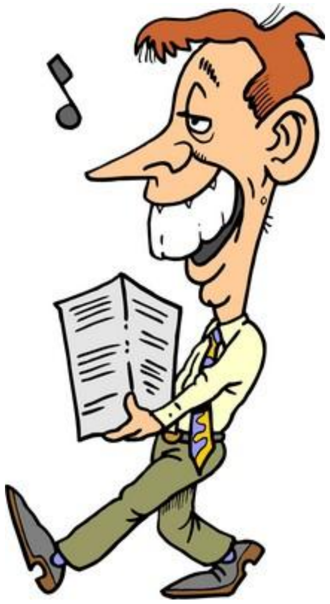
Maintains an intention *until* it believes the intention has been *achieved*

Aka *fanatical* commitment

Single-minded commitment

Maintains an intention *until* it believes that either:

- the intention has been *achieved*
- is *no longer possible* to achieve



Open-minded commitment

Maintains an intention *as long as*

- it has not been *achieved*
- it is still *desired*

Outline

- Introduction to agent architectures
- Abstract architectures for agents
- Deductive reasoning agents
- Agents as intentional systems
- **Reactive agents**
- Hybrid architectures

Reactive agents

- Many **problems** with **symbolic/logical approaches**, for example:
  - Inherent **computational complexity** of theorem proving
  - Cannot **operate effectively in time-constrained environments**

Reactive agents

- Many **problems** with **symbolic/logical approaches**, for example:
  - The **environment cannot change** while the **agent is making a decision**
  - **Not easy to represent and reason** about complex and dynamic environments

Reactive agents

- In the mid to late 1980s, researchers started to **investigate alternatives to symbolic AI paradigm**

- These **new approaches** had a few themes in common:

    - **Rejection of symbolic representation** and syntactic manipulation (e.g., logic programming)

Reactive agents

- These **new approaches** had a few themes in common:
  - The idea that **intelligent behavior** is **linked to the environment**
  - **Intelligent behavior can emerge** from the interaction of various simpler behaviors

Reactive agents

What are reactive agents?

- **Agents equipped with simple processing units** that perceive and quickly **react to changes** in the environment
- **Do not use complex symbolic reasoning**

Reactive agents

What are reactive agents?

- In reactive agent systems, **intelligence** is **not a property of a single component**
- The **intelligence is distributed** in the system and **emerges from the interaction** among agent components and the environment

Purely reactive agents

- **Purely reactive agents** make **no reference to their history**
  - no internal state
  - decision making entirely on the present

  - Formally:

$$Ag: E \rightarrow Ac$$

```
function decide(perception)
  current_state <- INTERPRET-INPUT(perception)
  rule <- RULE-MATCH(current_state,rules)
  action <- RULE-ACTION(rule)
return action
```

(Russell and Norvig, 1994)

Reactive architectures

Inspiration: **intelligent behavior** of animals in the world

- **simple behaviors** of each **individual agent**
- **complex behaviors** comes from **combining individual behaviors**

Brooks: subsumption architecture

Dr. Rodney Brooks’ short bio:

- 1981: **PhD Stanford**
- 1984 - 2010: Professor at **MIT**
- 1997 – 2007: Director of **MIT Artificial Intelligence Lab**

- 1990 - 2011: Founder, Board Member, and CTO of **iRobot Corp**
- 2008 – 2018: Founder, Board Member, and CTO of **Rethink Robotics**
- 2019 - Present: Founder and CTO **of Robust.AI**

Dr. Brooks was one of the most vocal and influential critics of the symbolic approach

Brooks: subsumption architecture

Decision-making:

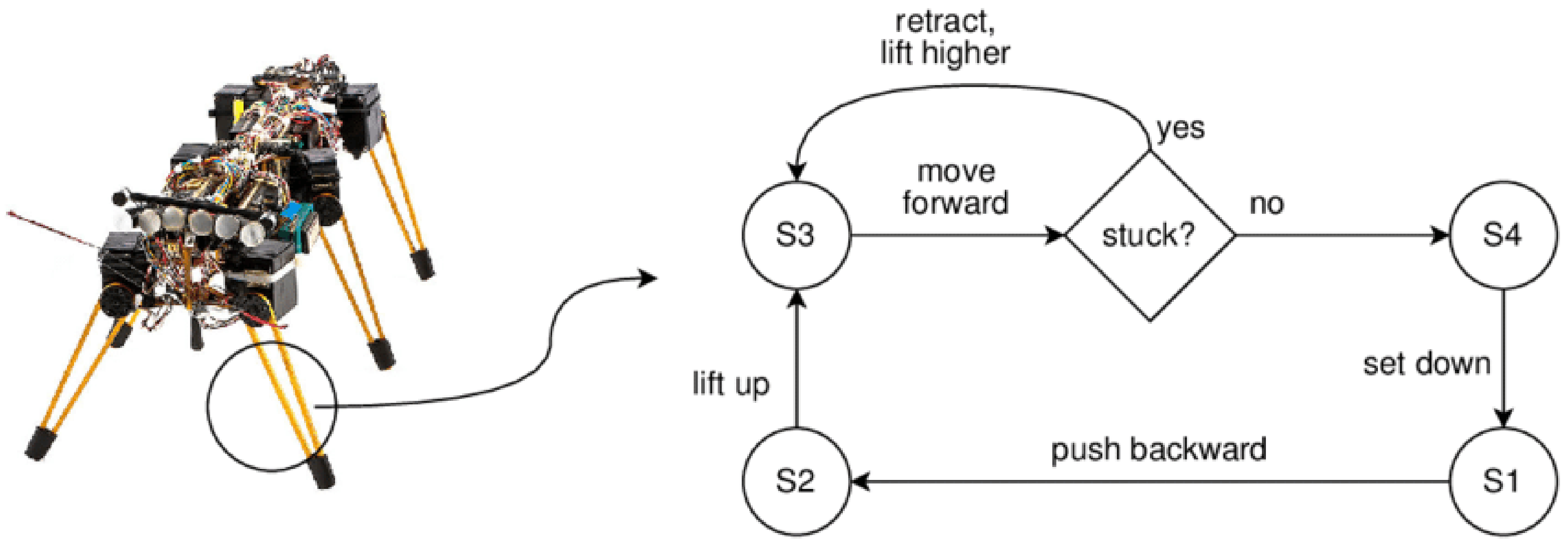
- A set of *task-accomplishing behaviors*
- Each **behavior module** can be seen as an **action selection function**

Brooks: subsumption architecture

- **Behavior module:**
  - **Perceptual inputs** are mapped into **actions**
  - Each **behavior module** is intended to **perform a task**

Brooks: subsumption architecture

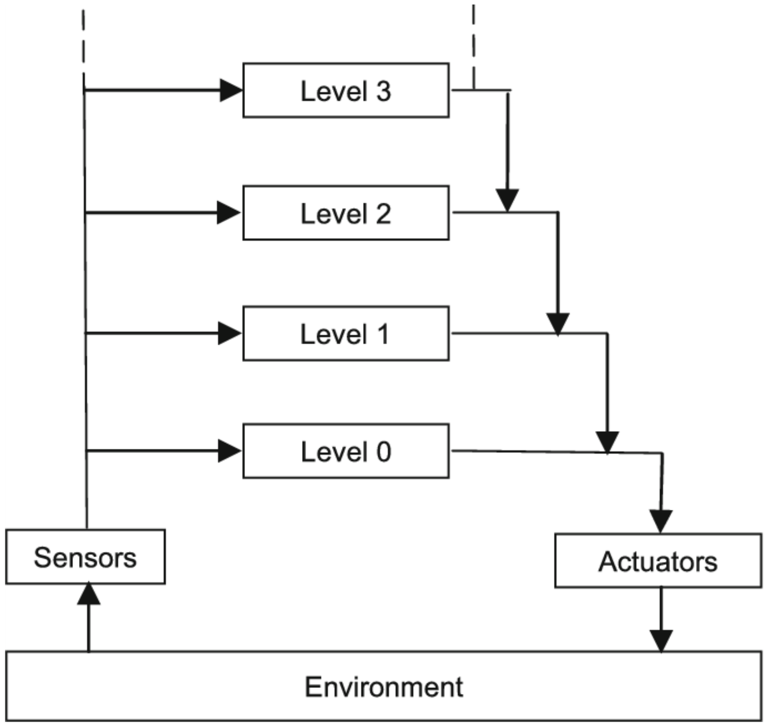
- Behavior modules are finite-state machines



Subsumption architecture: layered control

- However, many behaviors can *fire simultaneously*

- Subsumption hierarchy:
  - behaviors arranged into layers
  - lower layers inhibit higher layers



Agent development using Brooks' architecture

1. **Create a module for a particular task**
   - should **link perception and action**
   - should work by itself

2. **Add more modules**
   - the **priorities** for the behaviors need to be **re-adjusted** every time **one module is added**

Agent development using Brooks' architecture

Common requirements:

1. Deal with **multiple goals**

2. Deal with **multiple sensors**
   - may provide conflicting data

3. Be **robust** in dealing with changes in the environment

4. Deal with **time constraints**

Advantages of reactive agents

- Simplicity
- Economy
- Computational tractability

Advantages of reactive agents

- Robustness against failure
- Elegance
- Extensibility

Outline

- Introduction to agent architectures
- Abstract architectures for agents
- Deductive reasoning agents
- Agents as intentional systems
- Reactive agents
- **Hybrid architectures**

Limitations of reactive agents

- Decisions are only based on local information: agents have a ***short-term* view**
- Agents ***need sufficient local information*** to make decisions
- **No learning**: how to guarantee reactive rules evolve?
- Not trivial to ***engineer*** agents with complex behavior
- **Not easy to predict complex behavior** when agents have a *high number of layers*

Criticisms to deliberative agents

- **Deliberation** and **Planning** with **incomplete info** can be a problem
- **Speed of decisions can be slow** to deal with the real world
- Many architectures rely on a **symbolic approach**
- The need to be **grounded to the real world**

Hybrid agents

Many researchers argue that **neither a completely deliberative nor completely reactive approach is suitable**

⇒ *hybrid* systems to marry classical and alternative approaches

Hybrid architectures

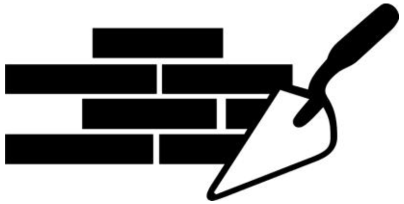
- Requirement: **agent must have both reactive and deliberative behaviors**
- Often, the **reactive subsystem is given some kind of precedence over the deliberative subsystem**
- This kind of structuring leads naturally to the **idea of a *layered* architecture**

Hybrid architectures

A key problem in hybrid architectures: **what kind of control flow should we consider between the agent's subsystems?**

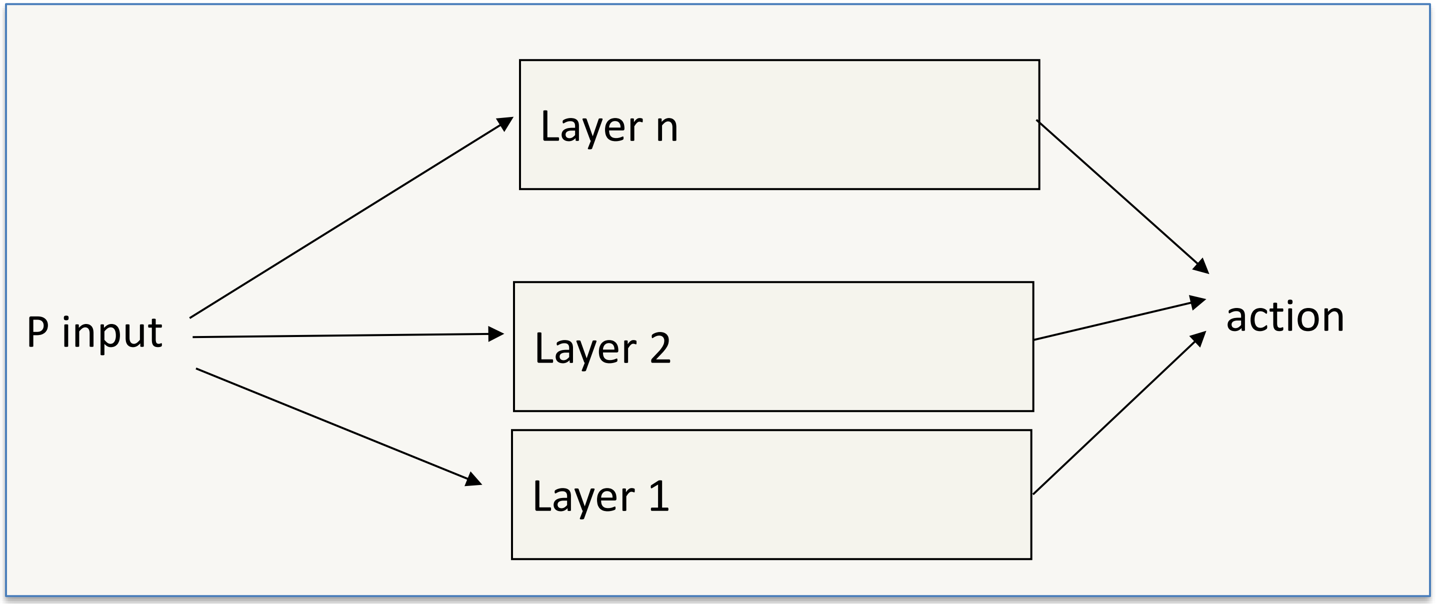
- *Horizontal layering*
  Layers are directly connected to the sensory input and action output. Each layer itself acts like an agent, producing suggestions on what action to perform

- *Vertical layering*
  Sensory input and action output are each dealt with by at most one layer each



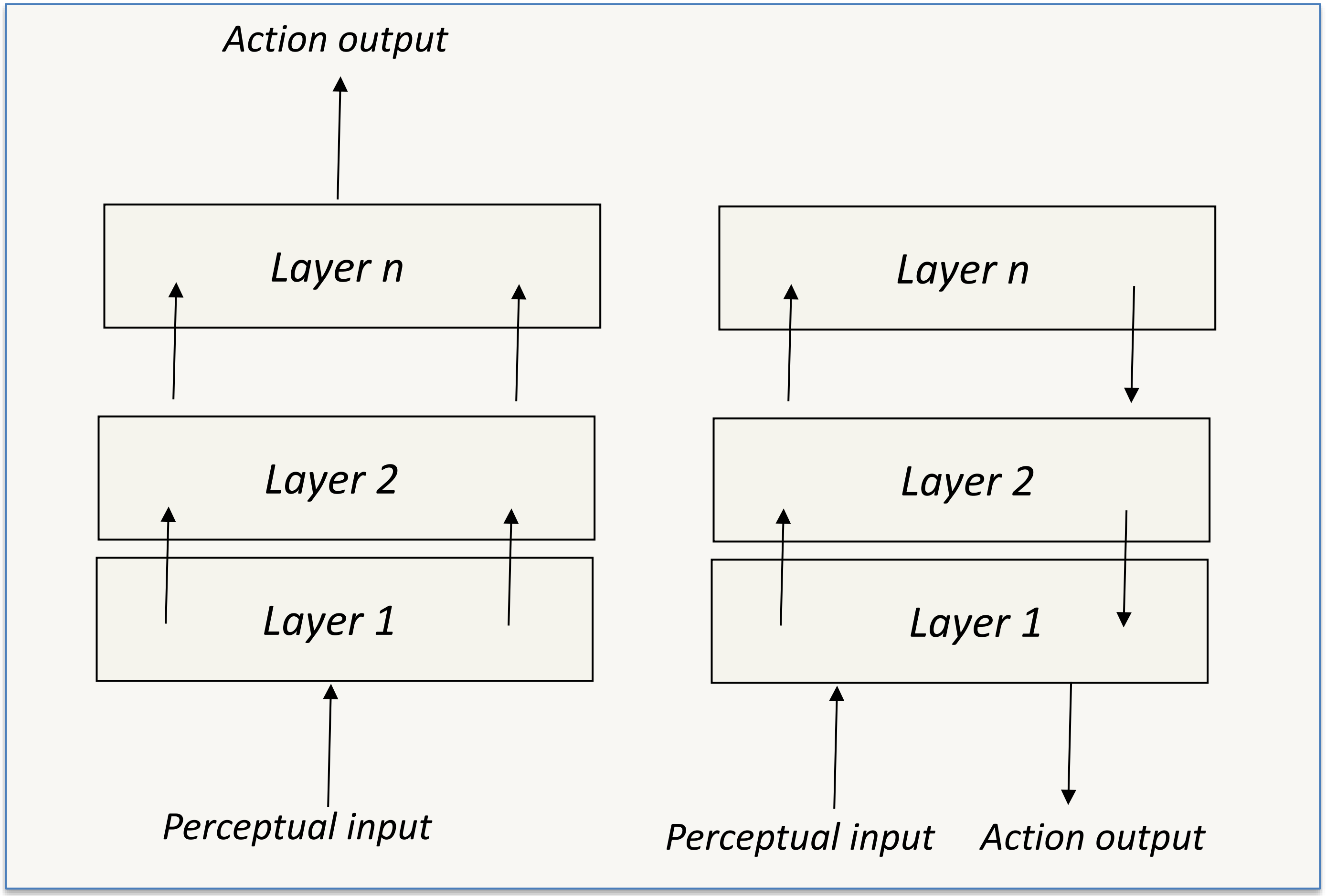
Hybrid architectures: horizontal layering

- **Advantages**: simple, distributed, fault-tolerant
- **Disadvantages**:
  - global behavior may not be coherent
  - difficult to avoid conflict between layers



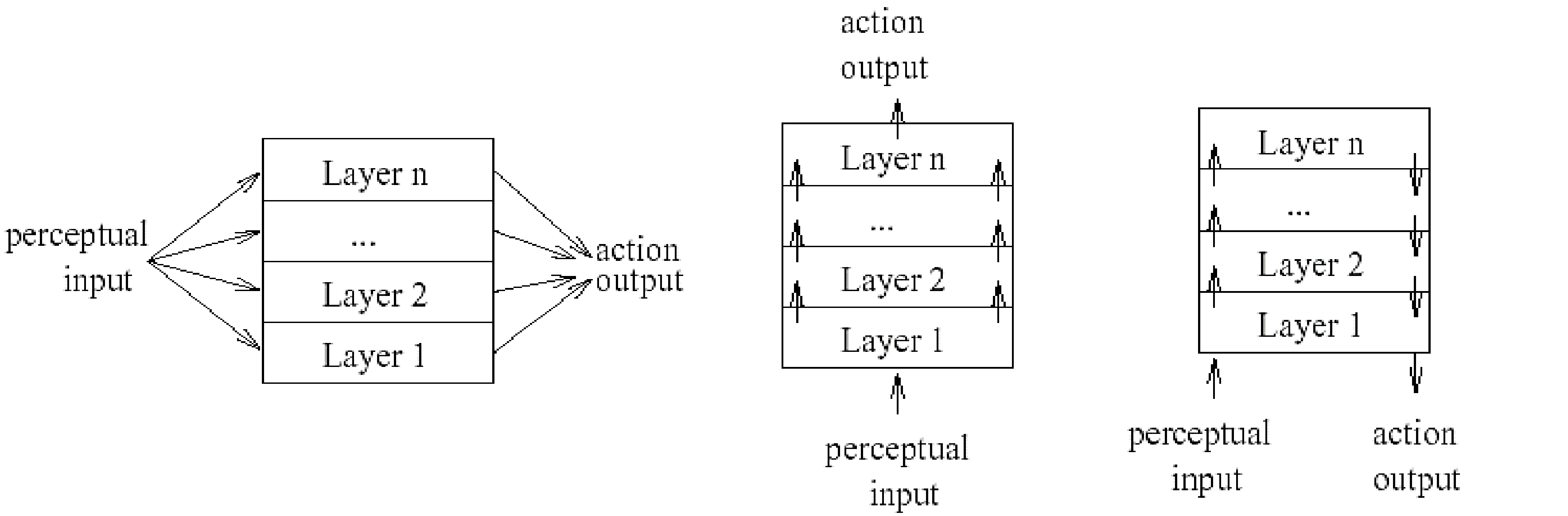
Hybrid architectures: vertical layering

Vertical (1 pass control) **Vertical** (2 passes control))



Comparing hybrid architectures

m possible actions suggested by each layer, *n* layers



(a) Horizontal layering

(b) Vertical layering (One pass control)

(c) Vertical layering (Two pass control)

complex decision making regarding the action output

not tolerant to layer failure

DARPA grand challenge[1]

Main goal of the challenge:

- develop an autonomous car (i.e., self-driving car):
    - capable of **traversing unrehearsed off-road terrain**
    - travel a **175 mile** long course through the **Mojave desert**
    - **take no more than 10 hours**

[1]https://en.wikipedia.org/wiki/DARPA_Grand_Challenge

DARPA grand challenge

- Teams
  - 2004 (1M$ prize): 107 teams (15 finalists, 0 finished)
  - 2005 (2M$ prize): 195 teams (23 finalists, 5 finished)
- **The route is kept secret** until 2h before the race
  - At this time they received a route description in RDDF format

Stanley – 2005 Winner

Volkswagen R5 (4WD)

- treats autonomous navigation as a software problem

Stanley: sensory equipment

Perception: roof rack that houses

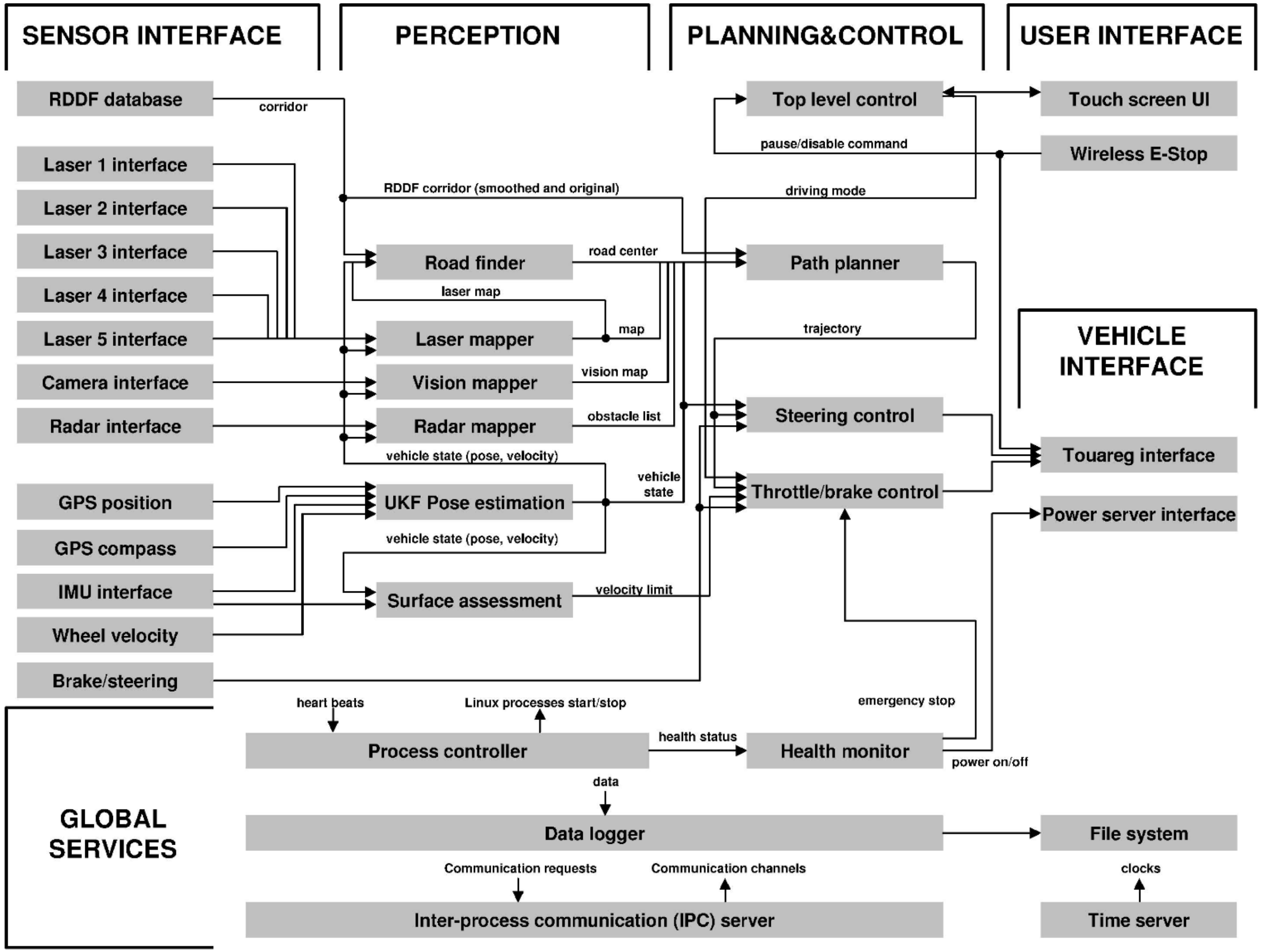
- 5 SICK laser range finders
- camera for long range perception
- 2 RADAR sensors
- antennae: 1 for GPS, 1 for GPS compass, and 1 radio antennae

Stanley control

3 main actuators

- brakes
- throttle
- steering

Stanley architecture



Stanley in action

http://www.youtube.com/watch?v=M2AcMnfzpNg

Since then…

Advantages of hybrid architectures

- One of the **most used architectures** currently
- Allows for a **real-time response** combined with **goal oriented-behavior**
- Reactivity can be privileged in relation to deliberation
- **Knowledge** about the world can be subdivided into layers
  - different **levels of abstraction**

Disadvantages of hybrid architectures

- **Vertical layering**: bottlenecks and fault intolerance
- **Horizontal layering**: complexity of decision making
- **Interactions between layers** are difficult to program and to test
  - one will need to analyze all the possible interactions between these layers
  - **an integration problem**

Thank You

rui.prada@tecnico.ulisboa.pt