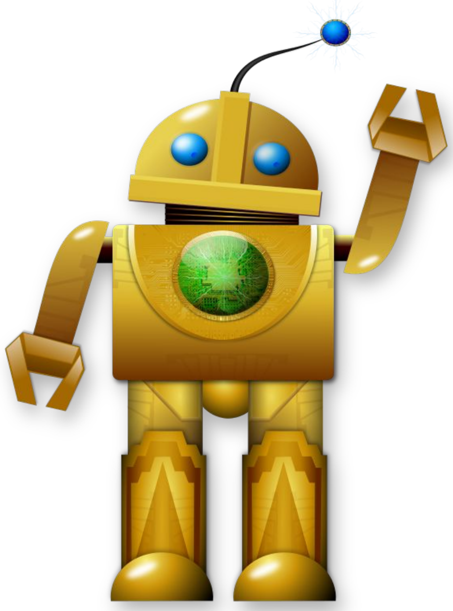
Intelligent Agents

Agents and environments
Rationality
PEAS
Environment types
Agent types



iClicker Question

How would you rate your knowledge of Python?

A. non-existent - what's Python, a snake?

B. basic - I can read and understand other people's code

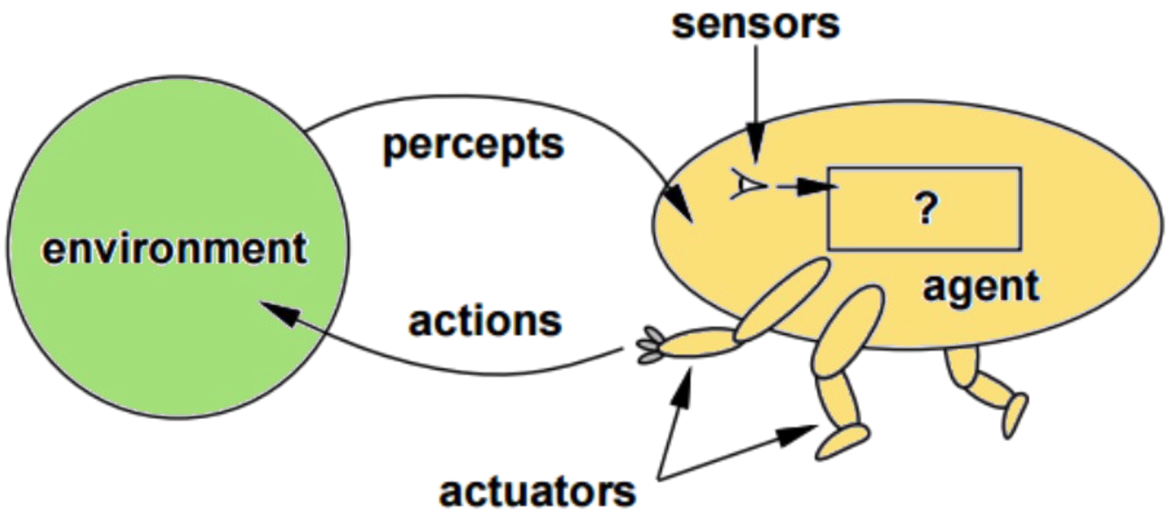
C. intermediate - I can write simple programs in Python

D. proficient - I am confident in my python programming abilities

E. expert - I can write a book on Python

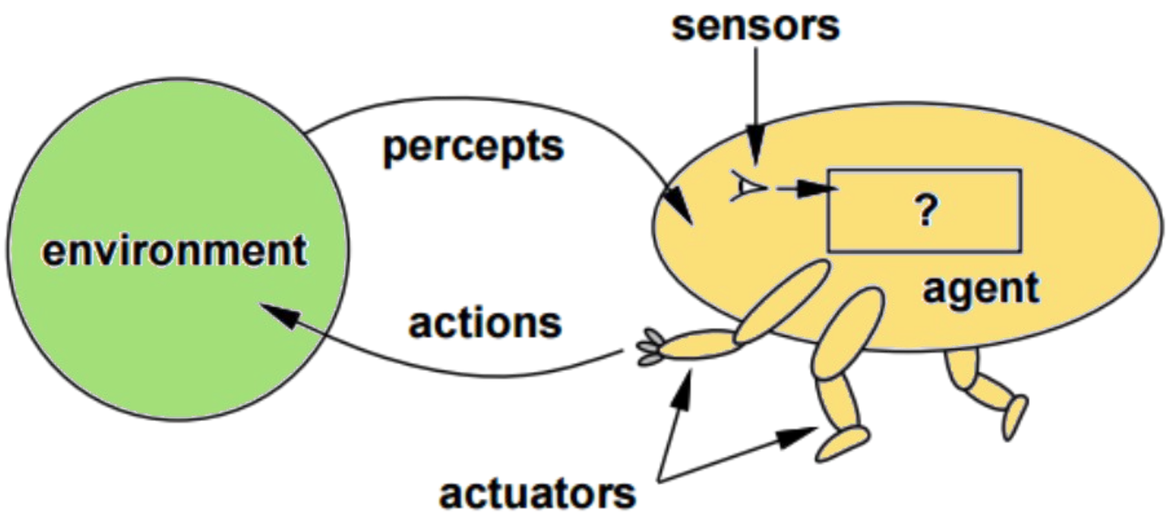
What is an Agent?

An agent is an entity that perceives its environment through sensors and acts upon its environment through actuators/effectors.



Agent vs Environment in AI

- We control the agent
- We have no direct control over the environment



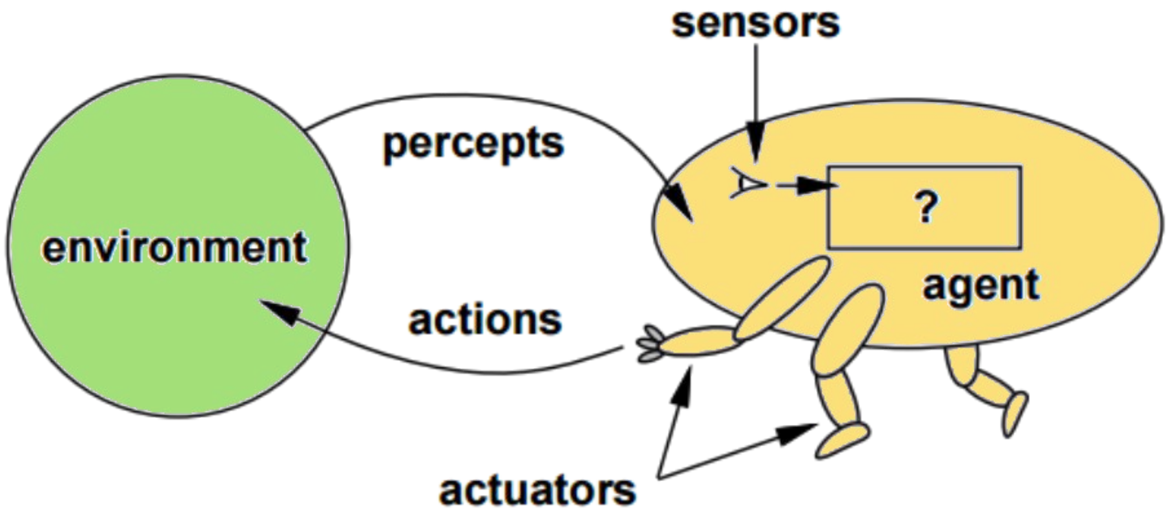
Human Agent

Sensors:

eyes, ears, nose, skin, tongue

Actuators:

hands/arms, legs, vocal cords



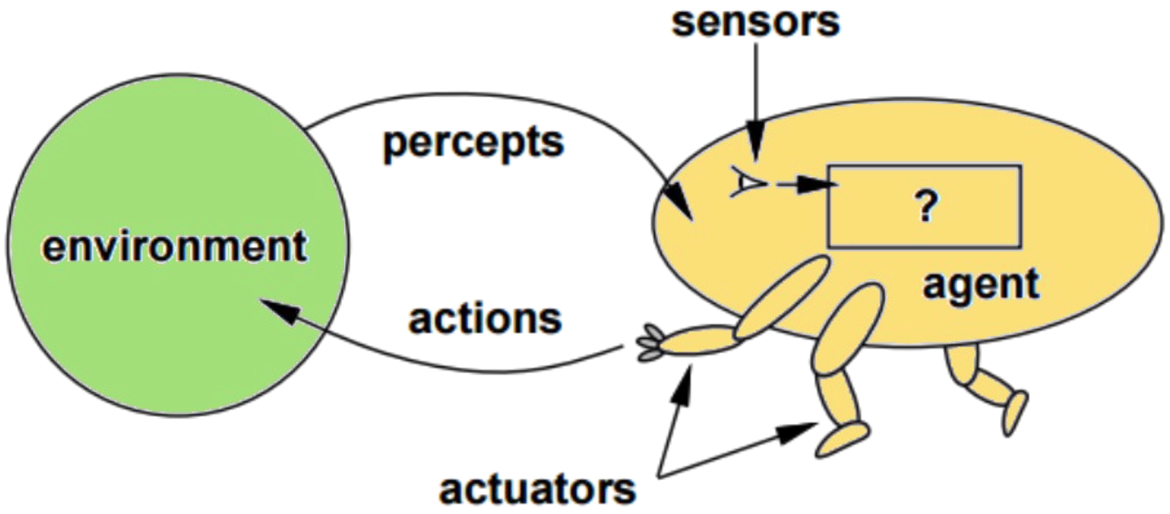
Robotic Agent

Sensors:

camera, microphone

Actuators:

motor, wheels



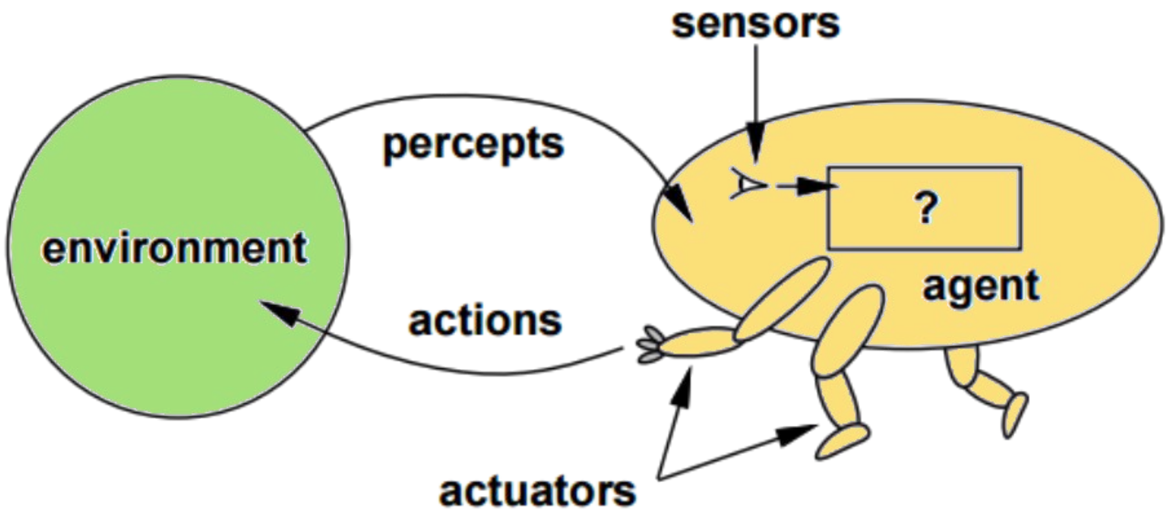
Software Agent

Sensors:

keyboard, files

Actuators:

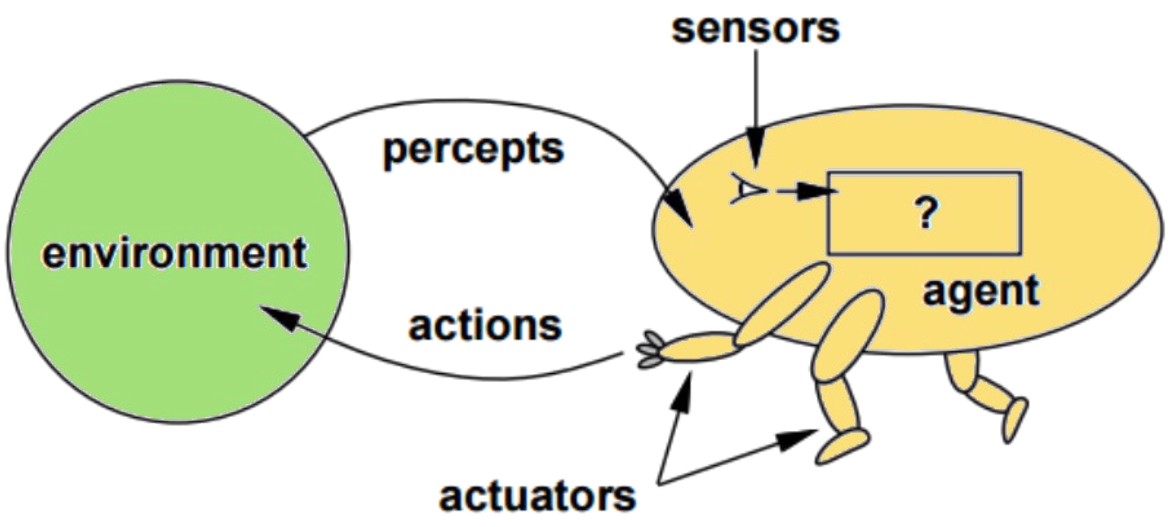
display, files



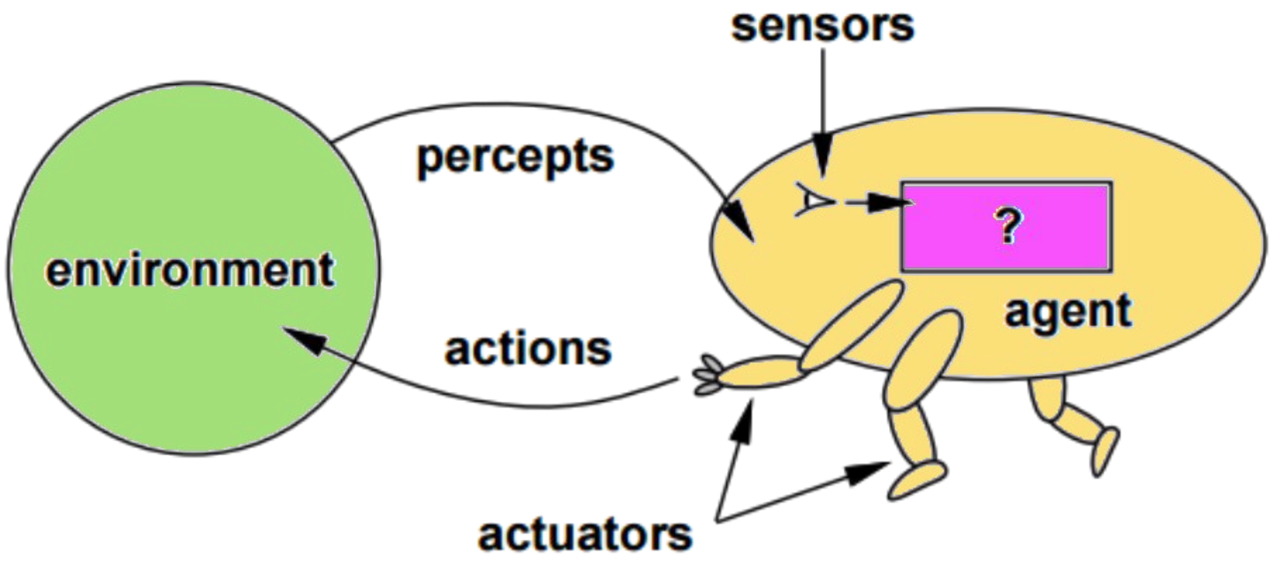
Percept

A percept is the agent's input at a given time.

The percept sequence is the complete history of everything the agent has ever perceived.



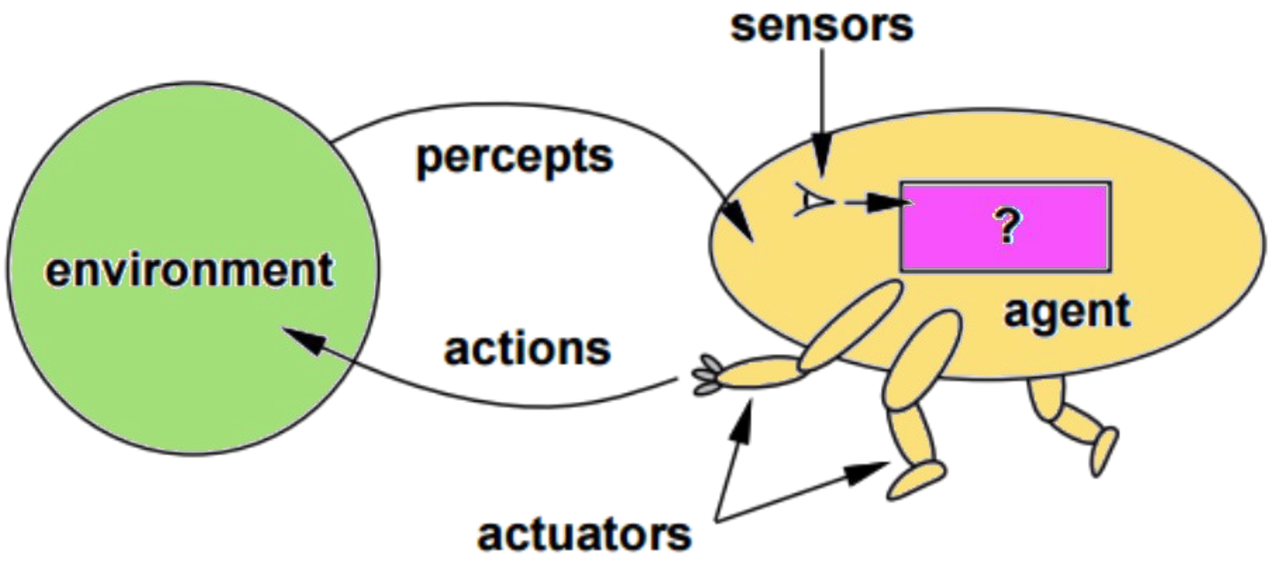
Agent Function



- The agent function maps percept histories to actions

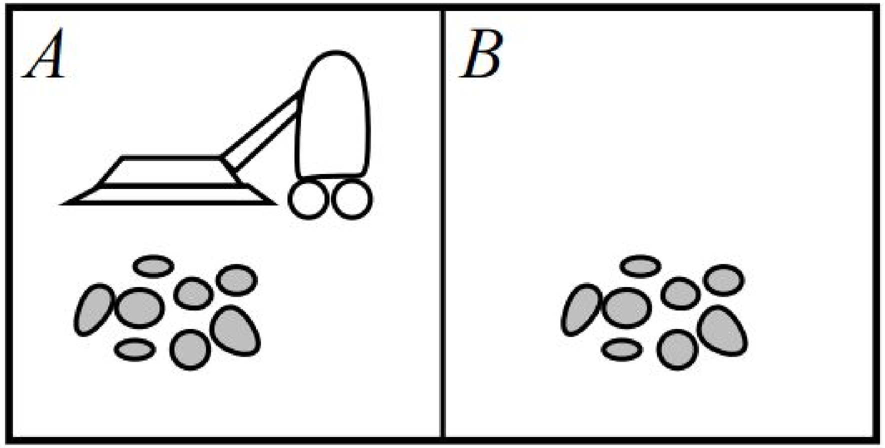
$f : P* \longrightarrow A$

Agent Program



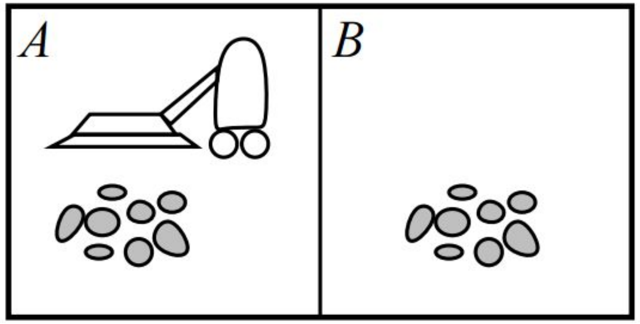
- The agent program runs on the physical architecture to produce the agent function
- agent = architecture + program

Vacuum Cleaner Agent



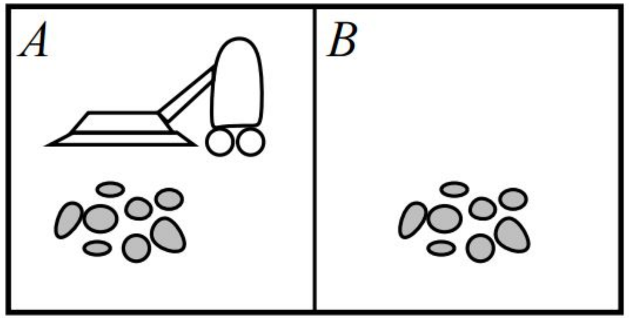
- Percepts: location and contents: [A, Dirty]
- Actions: Left, Right, Suck, NoOp

Vacuum Cleaner Agent Function



Percept Sequence	**Action**
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
[A, Clean], [A, Clean]	Right
[A, Clean], [A, Dirty]	Suck
….	…

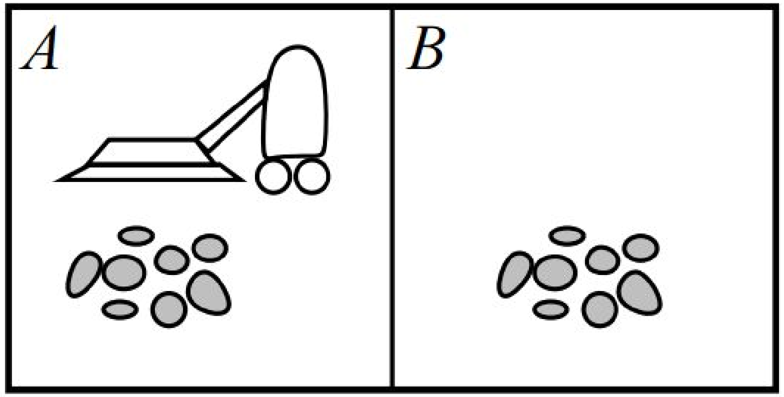
Vacuum Cleaner Agent Program



```
function reflex_vacuum_agent([location,status]) returns an action
    if status == Dirty then return Suck
    else if location == A then return Right
    else if location == B then return Left
```

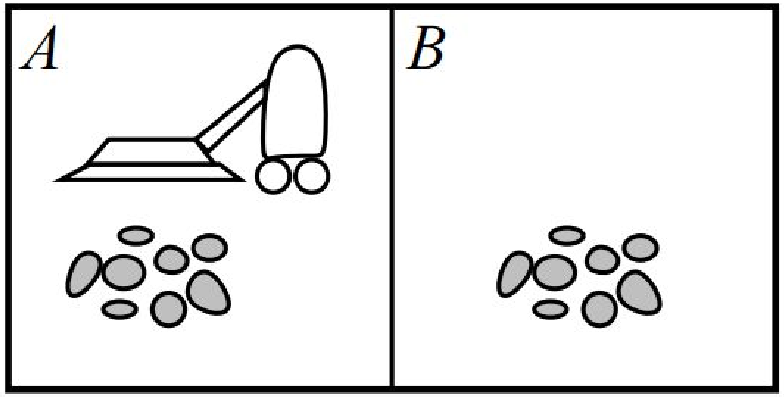
Vacuum Cleaner Agent Program

```
def reflex_vacuum_agent(location, status):
    '''
    Agent program for our vacuum cleaner
    :params
    location (A/B)
    status(Clean/Dirty)
    :return
    action - Left, Right, Suck
    '''
```



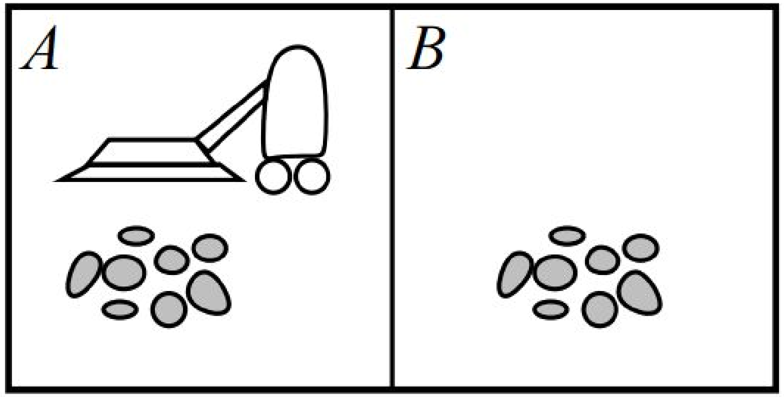
Vacuum Cleaner Agent Program

```python
def reflex_vacuum_agent(location, status):
    if status == "Dirty":
        return "Suck"
    elif location == "A":
        return "Right"
    else: # location is B
        return "Left"
```



Vacuum Cleaner Agent Program

```python
def reflex_vacuum_agent(location, status):
    '''
    Agent program for our vacuum cleaner
    :params
    location (A/B)
    status(Clean/Dirty)
    :return
    action - Left, Right, Suck
    '''
    if status == "Dirty":
        return "Suck"
    elif location == "A":
        return "Right"
    else: # location is B
        return "Left"
```



Rationality?

- Is our vacuum cleaner agent rational?
- Is it doing the 'right thing'?
- Is it successful?
- It depends

We need **a fixed performance measure** to figure out **if** and **how** successful an agent is.

Performance Measure

What is the criterion for success?

- amount of dirt cleaned? The vacuum will perform better if the dirt is sucked then dumped back?
- number of clean squares?
- penalty for movement?

The performance measure must be objective and defined in terms of **desired effect on the environment** (not on the actions of the agent)

Rational Agent

Rationality depends on:

- the performance measure
- the agent's knowledge of the environment
- the actions that the agent can perform
- the agent's percept sequence

A rational agent chooses the action that maximizes the expected value of the performance measure given the percept sequence to date and its knowledge of the environment.

Rationality - Limitations

- A rational agent is not omniscient: percepts may not supply all relevant information.
- A rational agent is not clairvoyant: percepts do not provide information about future events.
- A rational agent is not necessarily successful.

Rationality - Limitations

- Difficulty: computational limitations may make perfect rationality unachievable
- Design best program for given resources

Rationality

- Rationality involves exploration, learning and autonomy.

Task Environment - PEAS

To design a rational agent, we must specify the task environment:

- **P**erformance measure
- **E**nvironment
- **A**ctuators
- **S**ensors

Task Environment - PEAS
Automated Taxi

- Performance measure: safety, destination, profits, legality, comfort
- Environment: streets/freeways, traffic, pedestrians
- Actuators: steering, accelerator, brake, horn, speaker/display
- Sensors: video, accelerometers, gauges, engine sensors, keyboard/microphone, GPS

Task Environment - PEAS
Internet Shopping Agent

- Performance measure: price, quality, appropriateness, efficiency
- Environment: websites, vendors
- Actuators: display to user, follow URL, fill in form
- Sensors: keyboard, HTML pages (text, graphics, scripts)

Task Environment - PEAS
Part Picking Robot

- Performance measure: percentage of parts in correct bins
- Environment: conveyor belt with parts, bins
- Actuators: jointed arm and hand
- Sensors: camera

Environment Properties
Fully Observable vs Partially Observable

Fully observable: the agent's sensors give it access to the **complete (relevant) state of the environment** at each point in time.

- Chess?
- Automated taxi?
- The real world?

When the environment is partially observable, the agent needs to keep track of what it has seen of the environment so far.

Environment Properties
Deterministic vs Stochastic

Deterministic: the next state of the environment is completely determined by the **current state and the action executed by the agent.**

- Chess?
- Automated taxi?
- The real world?

Environment Properties
Episodic vs Sequential

Episodic: The agent's experience is divided into atomic episodes. Each episode consists of the agent perceiving and then performing a single action, **and the choice of action in each episode depends only on the episode itself.**

- Chess?
- Automated taxi?
- Part picking robot?
- The real world?

Environment Properties
Static vs Dynamic

Static: the environment does not change while an agent is deliberating.

Semidynamic: the environment itself does not change with the passage of time but the agent's performance score does.

- Chess?
- Automated taxi?
- The real world?

Environment Properties
Discrete vs Continuous

Discrete: a finite number of distinct, clearly defined percepts and actions.

- Chess?
- Automated taxi?
- The real world?

Environment Properties
Single Agent vs Multiagent

Single agent: an agent operating by itself in an environment.

Multiagent environments may be competitive or cooperative.

- Chess?
- Automated taxi?
- Part picking robot?
- The real world?

Environment Properties

Consider an agent playing checkers.

Which of the following is NOT a property of its environment?

A. fully observable
B. deterministic
C. episodic
D. static
E. discrete

Environment Properties

Consider an agent playing backgammon.

Which of the following is a property of its environment?

A. partially observable
B. deterministic
C. episodic
D. discrete
E. single-agent

Environment Properties

Consider an agent performing medical diagnosis.

Which of the following is a property of its environment?

A. fully observable
B. deterministic
C. episodic
D. dynamic
E. discrete

Environment Properties

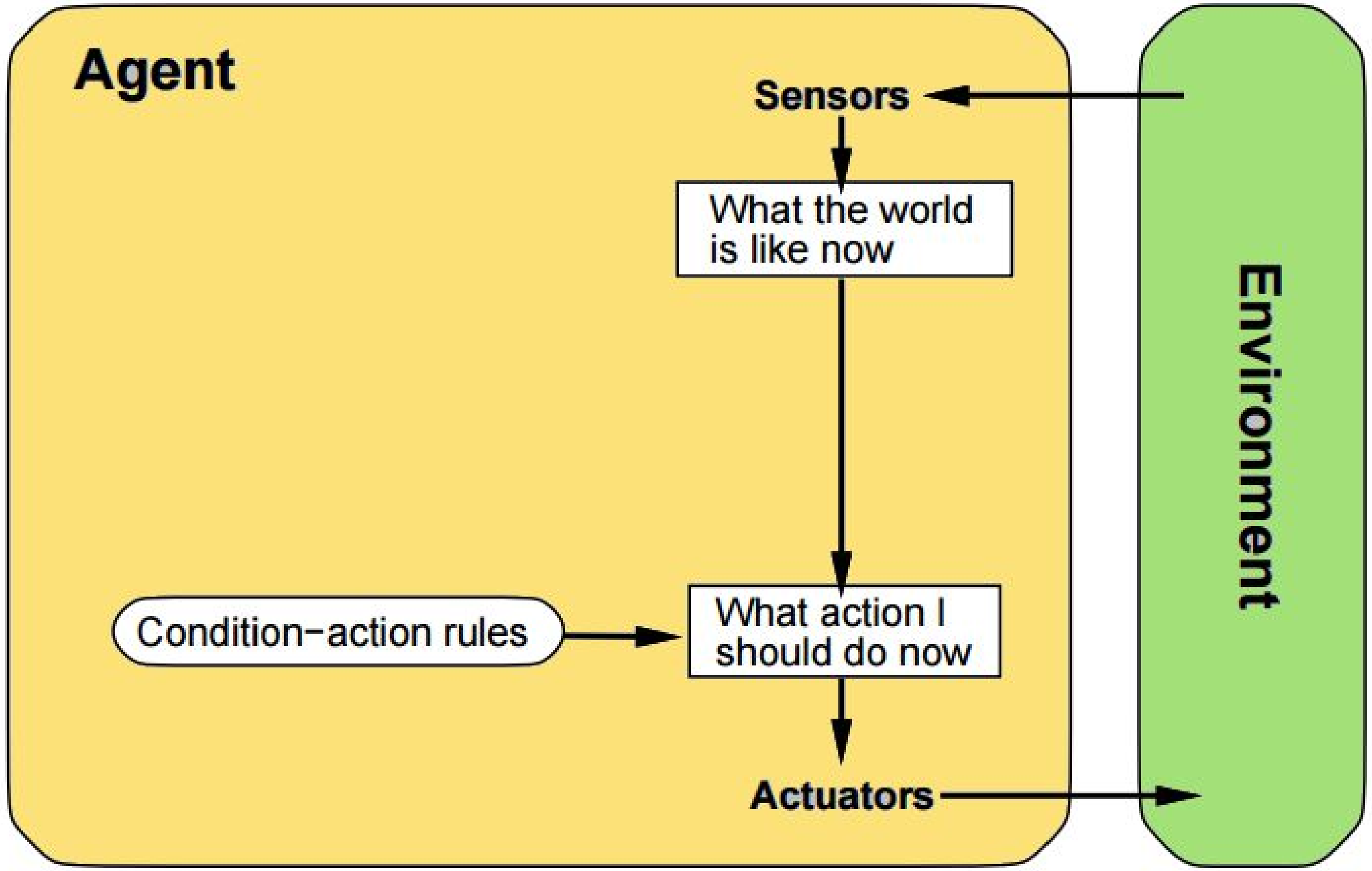
	Checkers	Backgammon	Medical Diagnosis
observable	Yes	Yes	No
deterministic	Yes	No	No
episodic	No	No	No
static	Yes	Yes	No
discrete	Yes	Yes	No
single agent	No	No	Yes

Agent Types

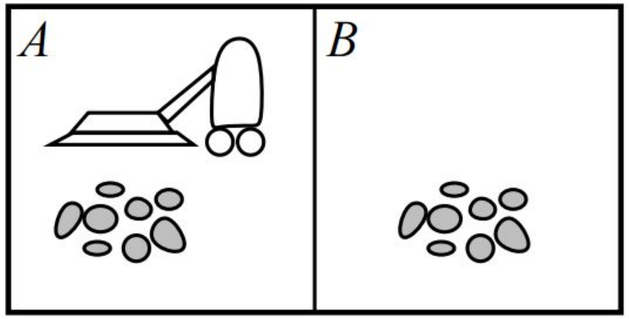
- simple reflex agents
- reflex agents with state
- goal-based agents
- utility-based agents

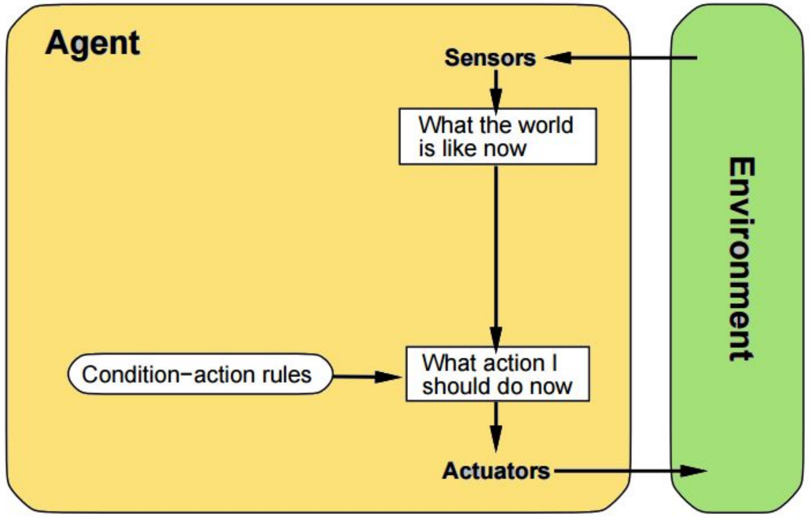
All these agents can be turned into learning agents.

Simple Reflex Agent



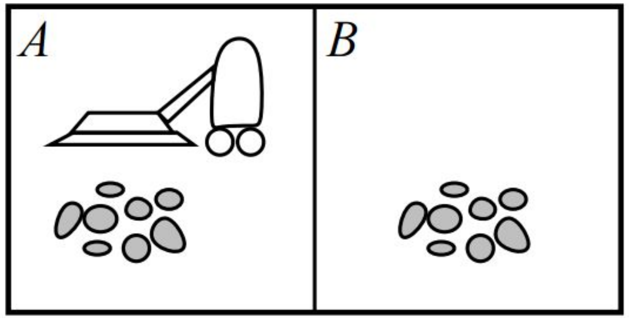
Simple Reflex Agent

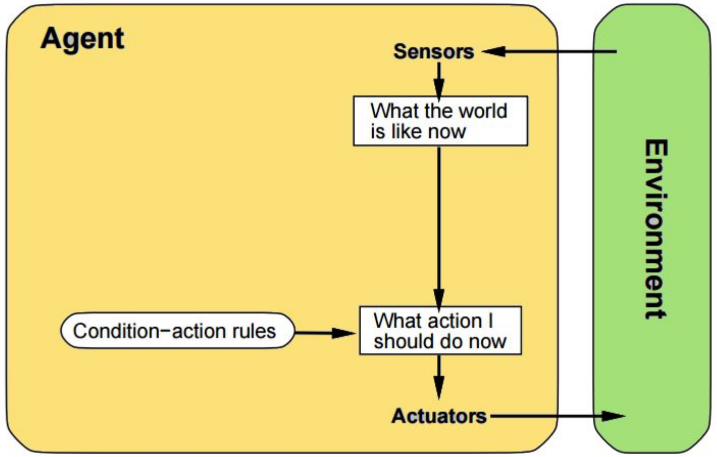




```
function reflex_vacuum_agent([location,status]) returns an action
    if status == Dirty then return Suck
    else if location == A then return Right
    else if location == B then return Left
```

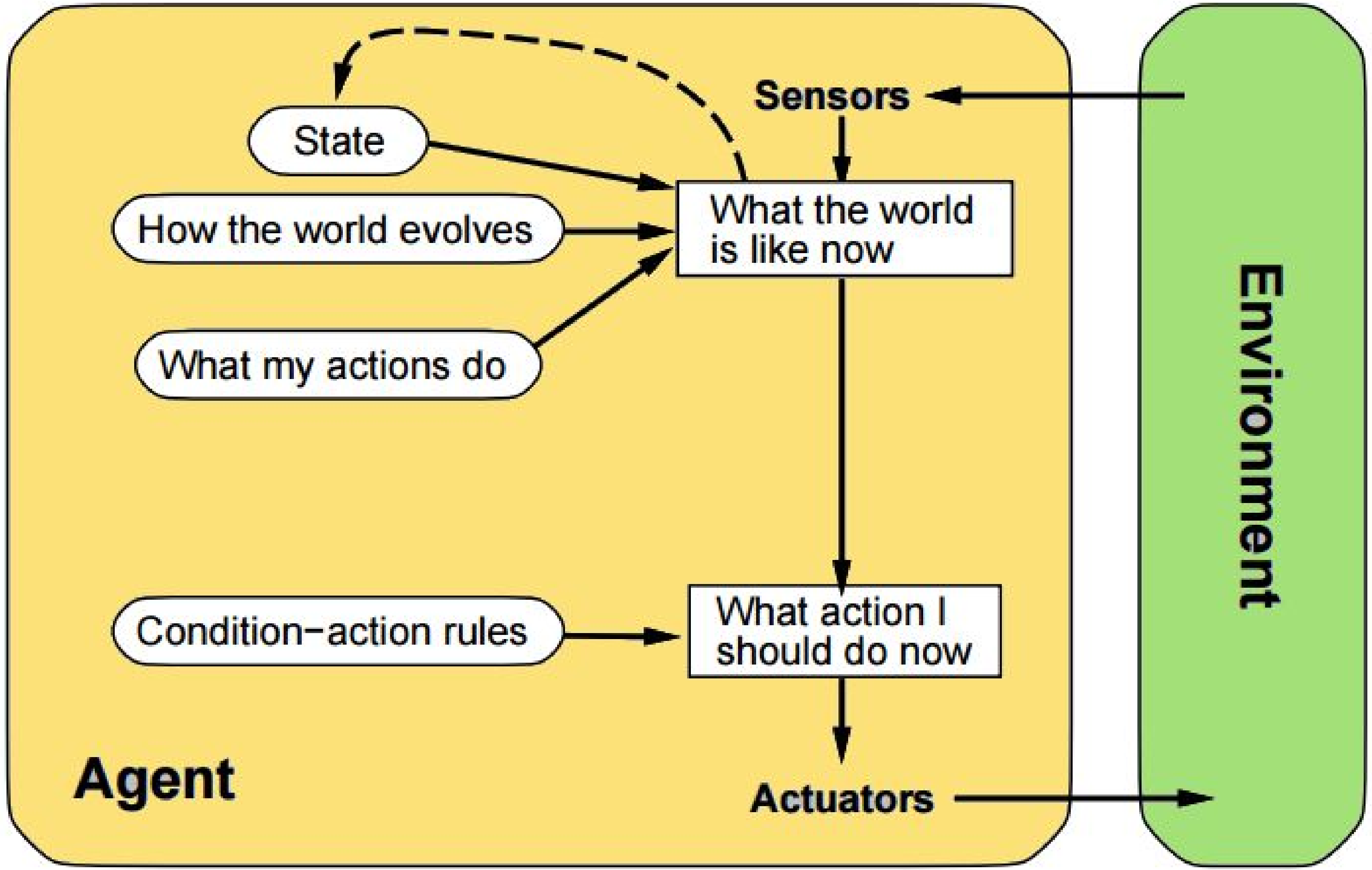
Simple Reflex Agent



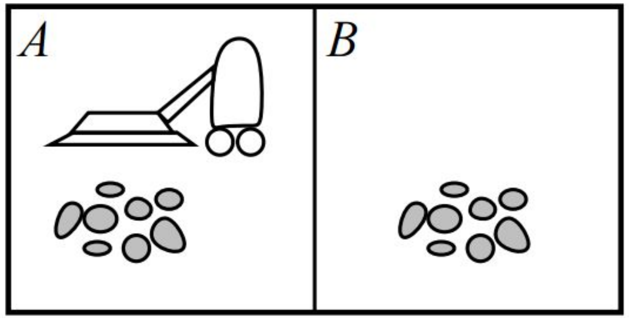


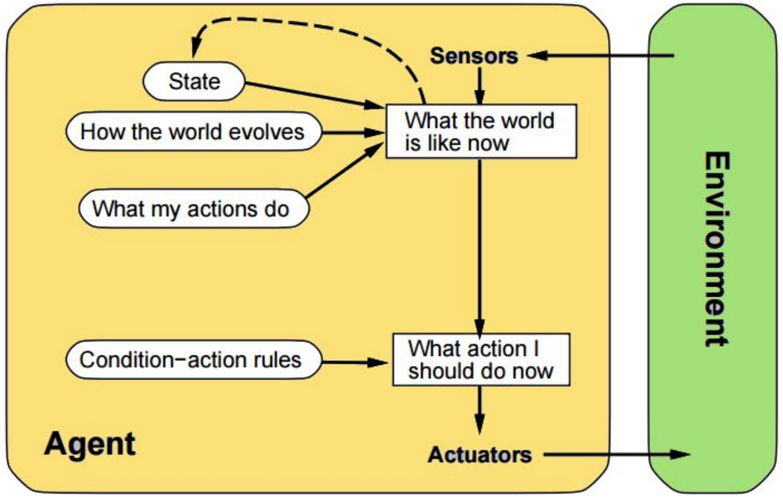
- ➢ Choose action based on current percept
- ➢ Do not consider the future consequences of its actions
- ➢ Consider how the world IS

Reflex Agents with State



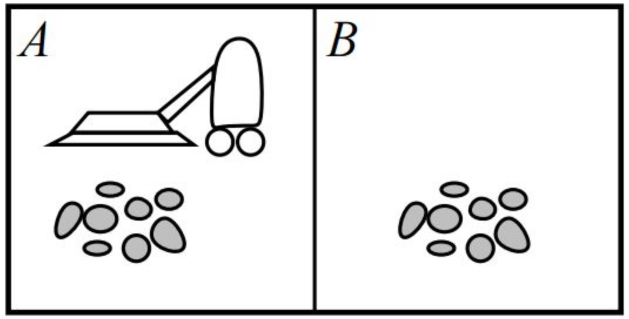
Reflex Agents with State

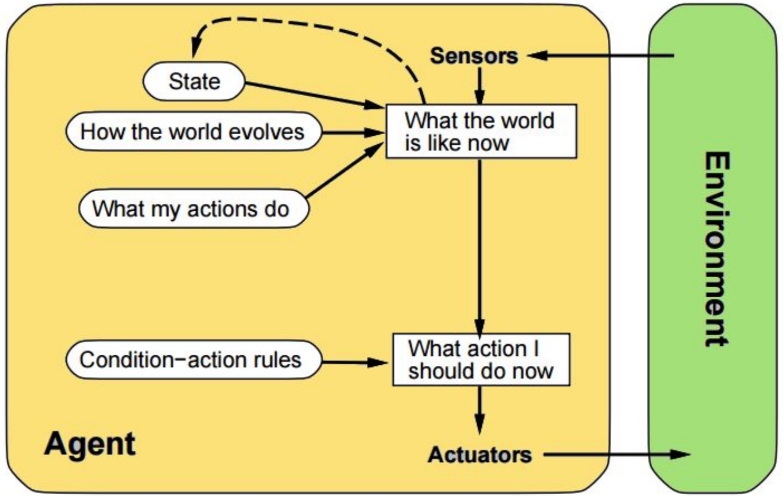




- ➢ Choose action based on current percept and memory
- ➢ May have memory or a model of the world’s current state
- ➢ Do not consider the future consequences of their actions
- ➢ Consider how the world IS

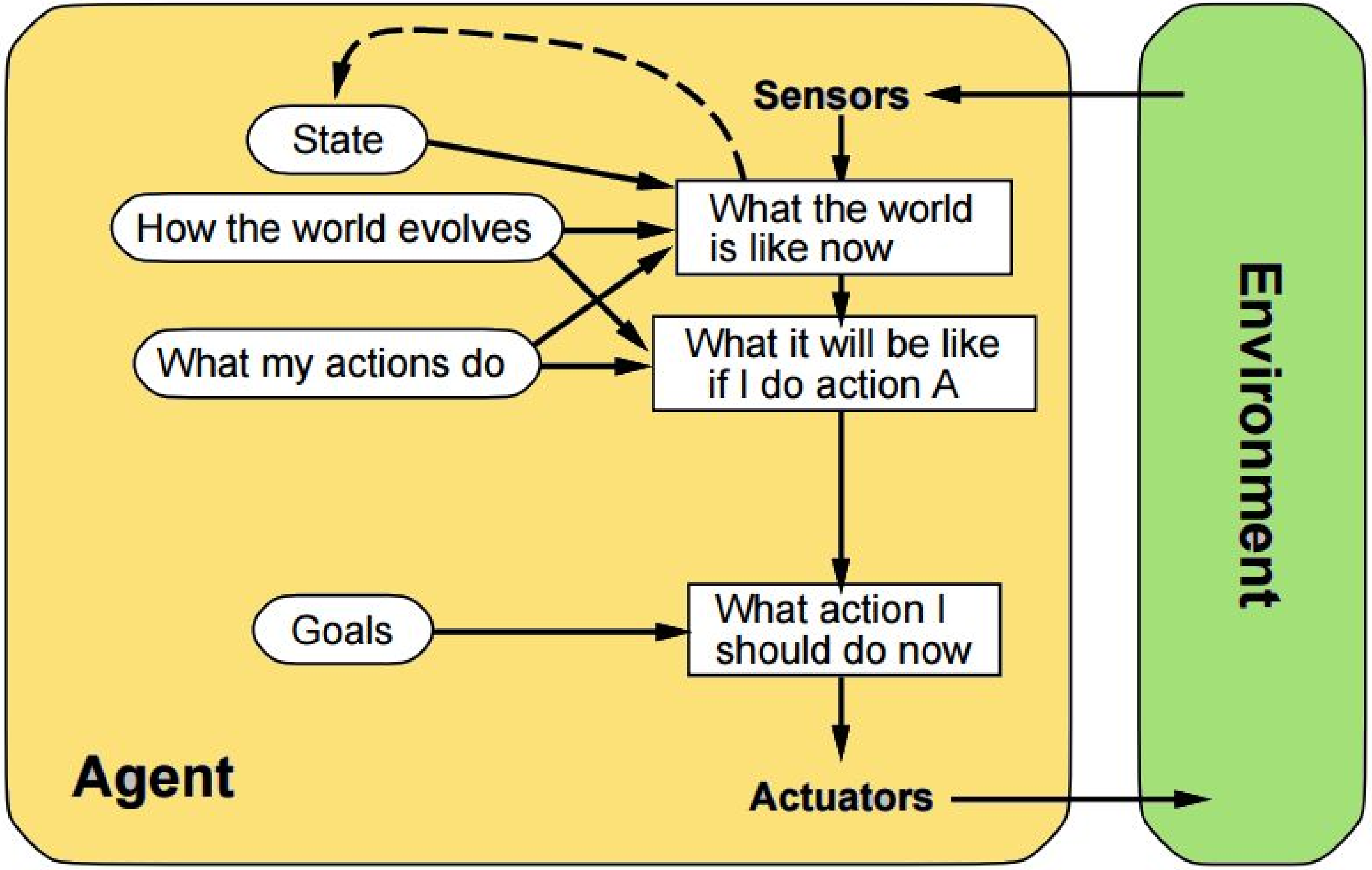
Reflex Agents with State



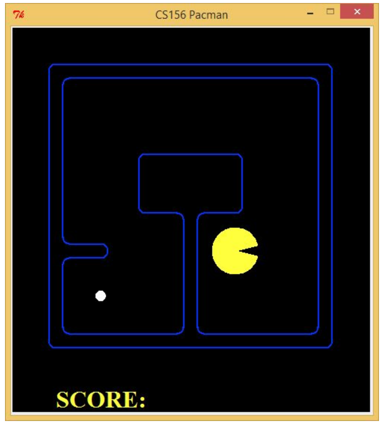


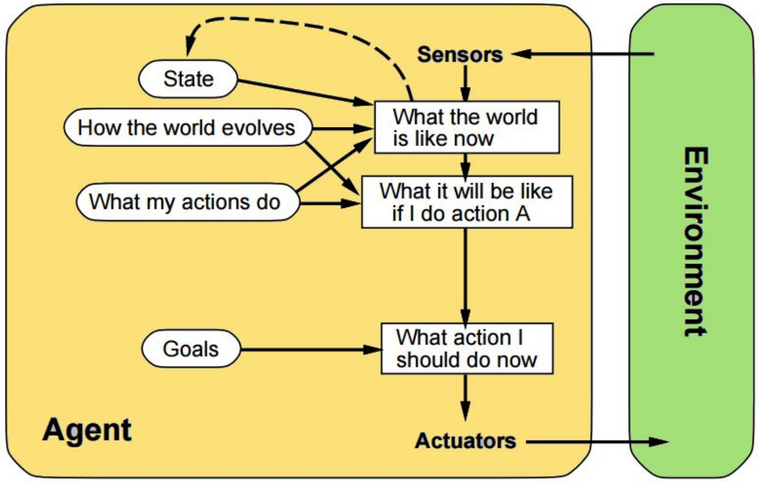
- ➢ Vacuum agent remembers the locations visited
- ➢ Suppose it knows that a clean location stays clean for some time t
- ➢ Does not go back to that location for a while

Goal Based Agents


Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

Goal Based Agents

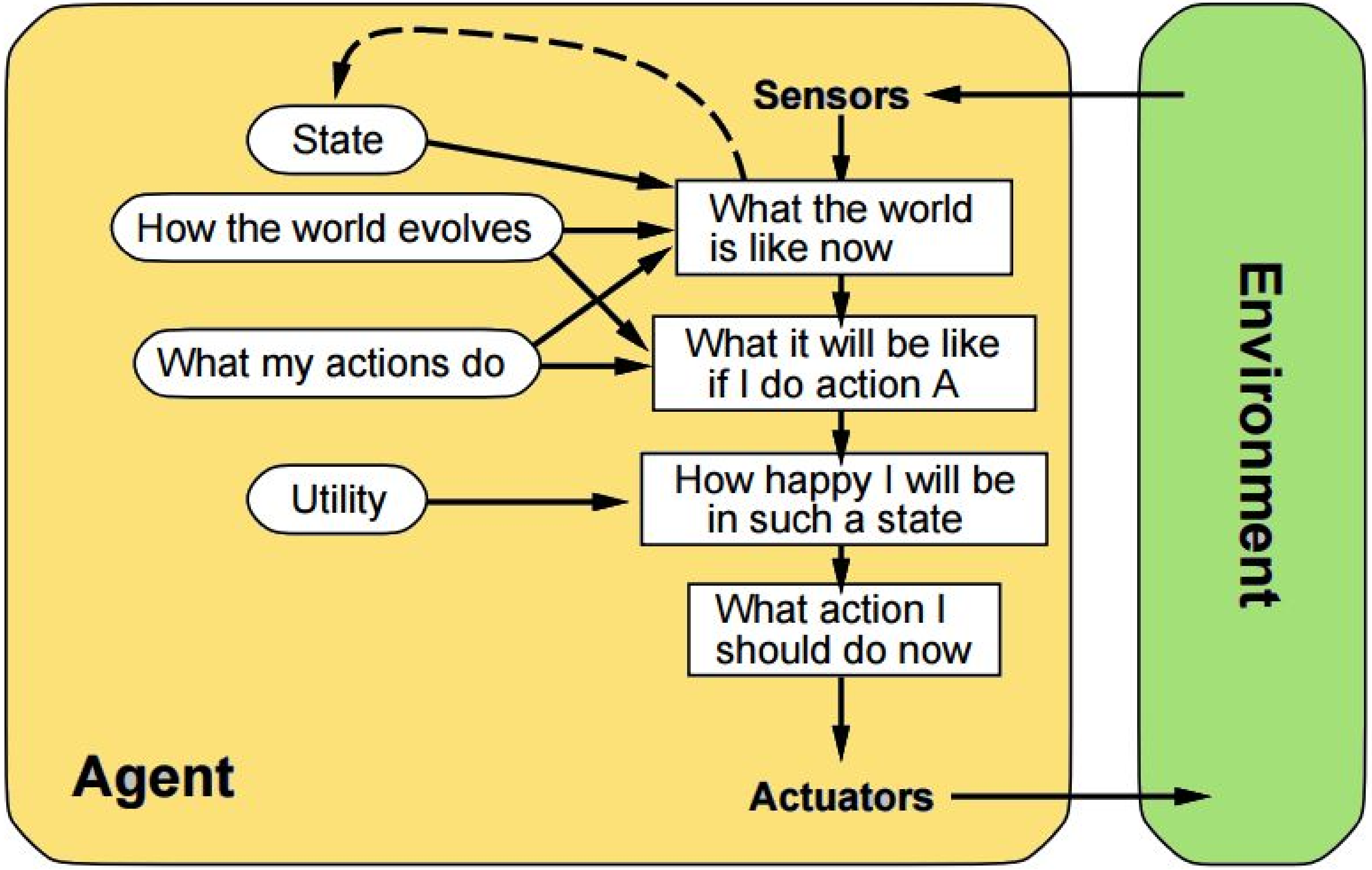




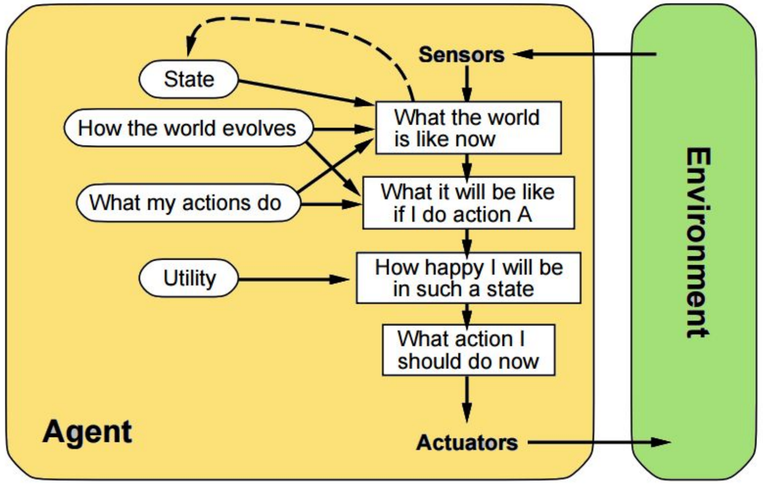
Sensors
State
What the world is like now
How the world evolves
What my actions do
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

- Ask "what if"
- Decisions based on (hypothesized) consequences of actions
- Must have a model of how the world evolves in response to actions
- Must formulate a goal (test)
- Consider how the world WOULD BE

Utility Based Agents


Sensors
State
How the world evolves
What my actions do
Utility
What the world is like now
What it will be like if I do action A
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

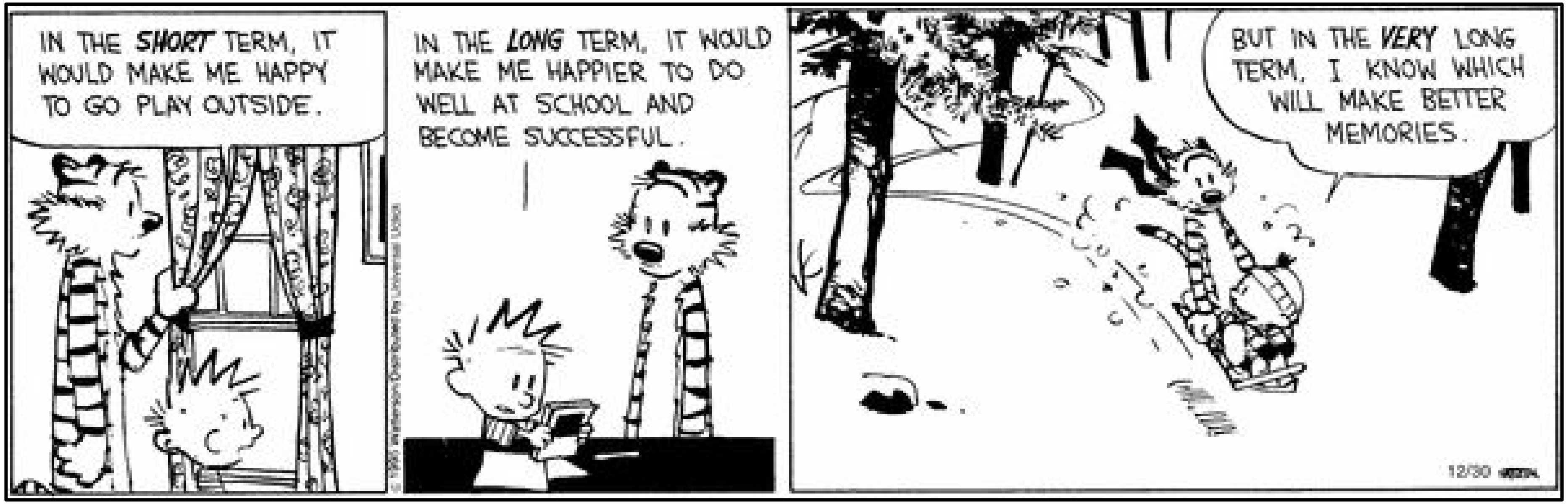
Utility Based Agents



Sensors
State
How the world evolves
What my actions do
Utility
What the world is like now
What it will be like if I do action A
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

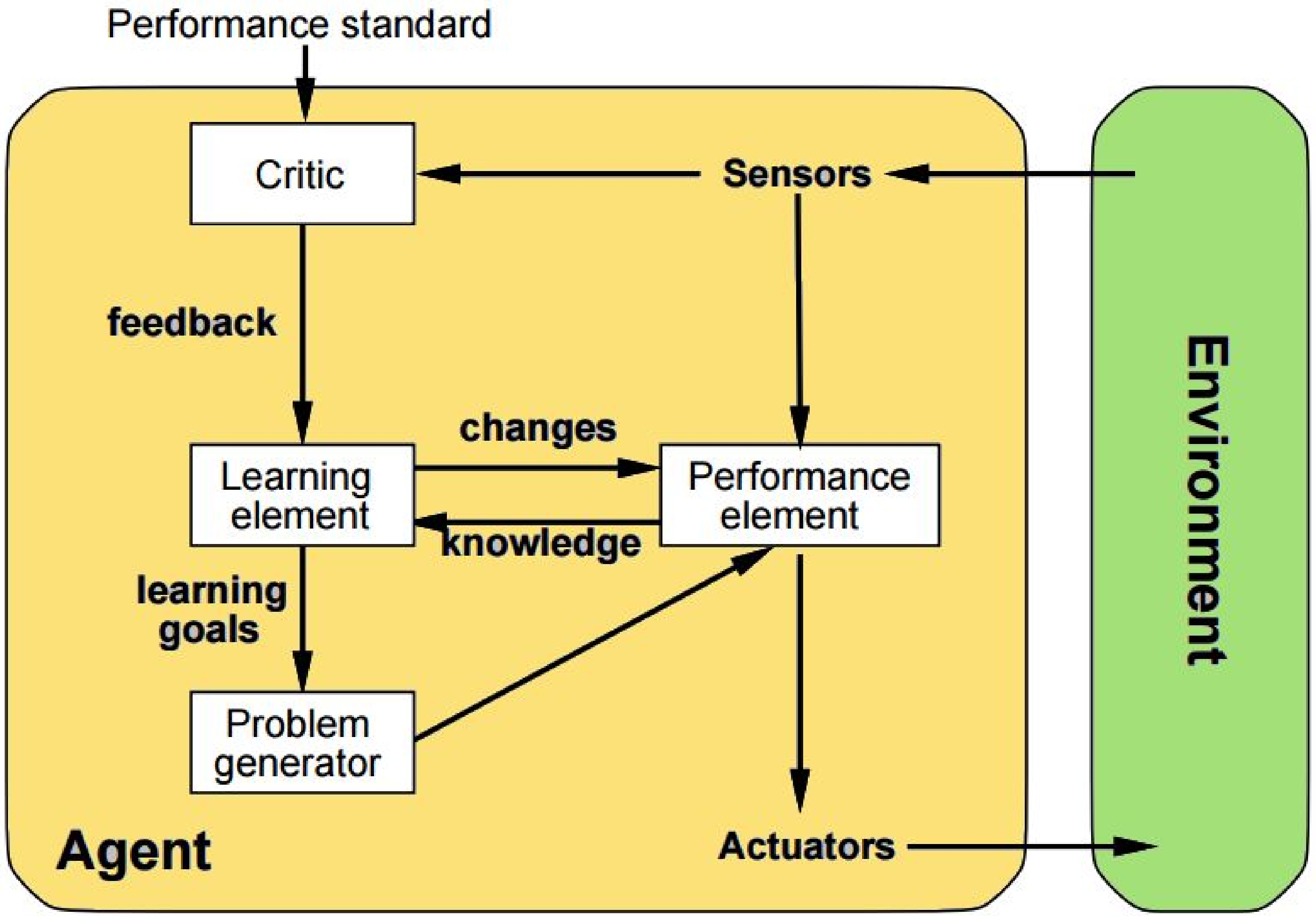
- ➢ Ask "what if"
- ➢ Decisions based on (hypothesized) consequences of actions
- ➢ Must have a model of how the world evolves in response to actions
- ➢ Must formulate a utility - a measure of 'happiness'
- ➢ Select actions that maximize the expected utility
- ➢ Consider how the world WOULD BE

Utility Based Agent?


IN THE *SHORT* TERM, IT WOULD MAKE ME HAPPY TO GO PLAY OUTSIDE.
IN THE *LONG* TERM, IT WOULD MAKE ME HAPPIER TO DO WELL AT SCHOOL AND BECOME SUCCESSFUL.
BUT IN THE *VERY* LONG TERM, I KNOW WHICH WILL MAKE BETTER MEMORIES.
12/30

Calvin and Hobbes by Bill Watterson

Learning Agents



Summary

- **An agent** perceives and acts in an **environment**
- The agent function describes what the agent does in all circumstances
- The **performance measure** evaluates the environment sequence
- A perfectly rational agent maximizes expected performance
- **PEAS** descriptions define task environments
- Environments are categorized along several dimensions: observable, deterministic, episodic, static, discrete, single-agent
- Basic agent architectures: reflex, reflex state, goal-based, utility-based