Course Topics

- **Intelligent agents (AIMA Ch. 2)**
- Search (AIMA Ch. 3, 4, 6, 5)
- Reasoning (AIMA Ch. 7-9, 12-15)
- Machine learning (AIMA Ch. 19-20)
- Deep learning (AIMA Ch. 22)
- Natural language processing (AIMA Ch. 24)
- Generative AI (Optional)

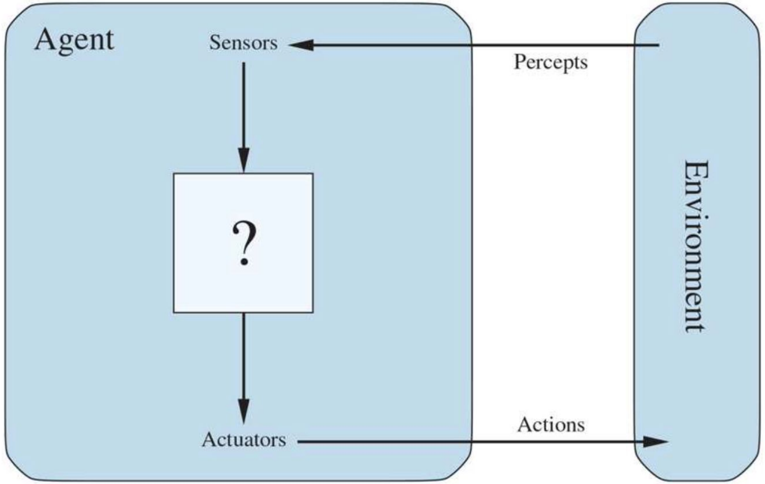
Intelligent Agent

L.-Y. Wei

Spring 2024

Agent

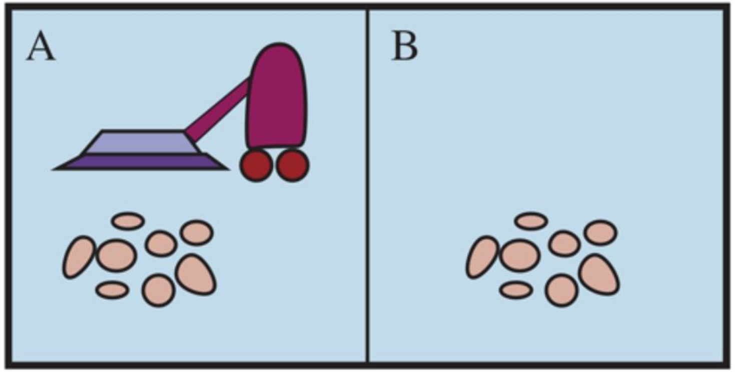
- An **agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**
    - Robotic agent
        - e.g., camera/robotic arm
    - Software agent
        - e.g., keyboard/writing files
    - …



- An agent's behavior is described by the **agent function** that maps any given percept sequence to an action f(percept sequence) = action

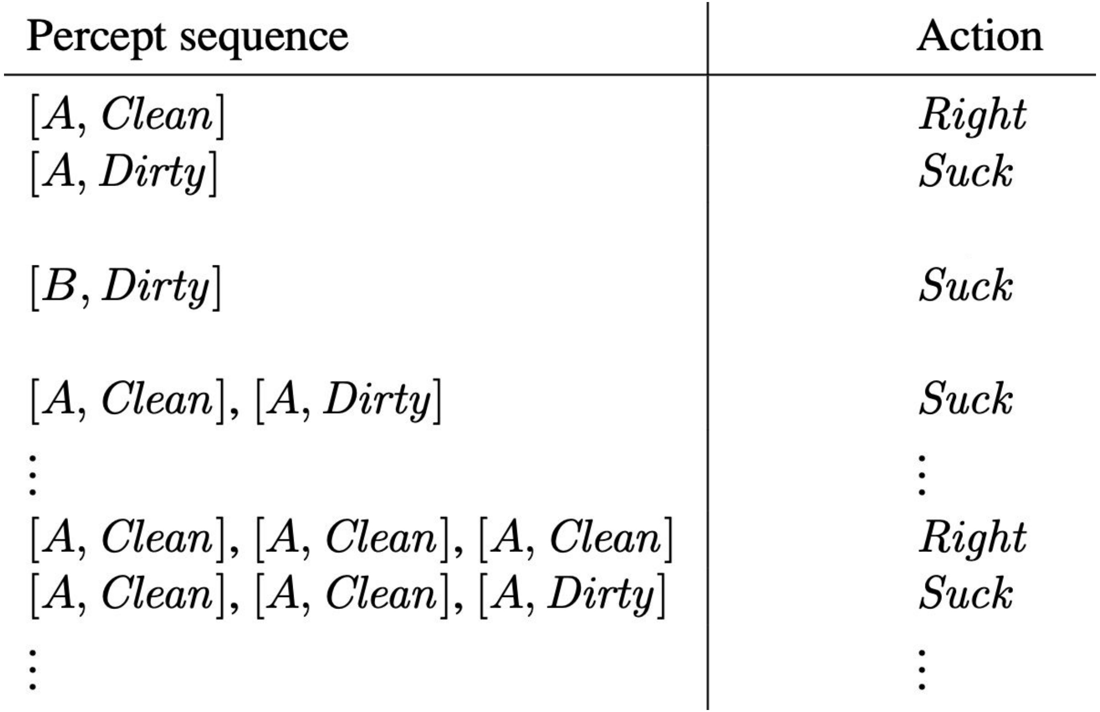
Example: Vacuum World

Source: Russell and Norvig, *AIMA, 2020*

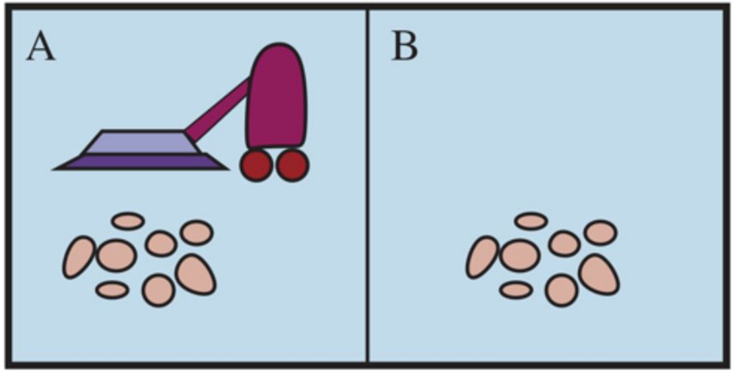


- Percepts
  - Location: A or B
  - Status: clean or dirty
- Actions
  - Move right
  - Move left
  - Suck (clean the square that it occupies)
  - NoOp

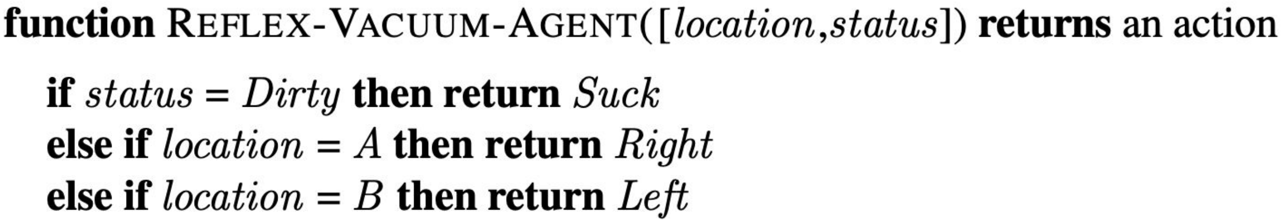
Source: Russell and Norvig, *AIMA, 2020*



Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Dirty]$	$Suck$
⋮	⋮
$[A, Clean], [A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Clean], [A, Dirty]$	$Suck$
⋮	⋮



What is the **right** function?



```
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
```

Rational Agent

- A rational **agent** should select an action that is **expected** to **maximize** its **performance measure**, given the percept sequence to date

* to date = until now

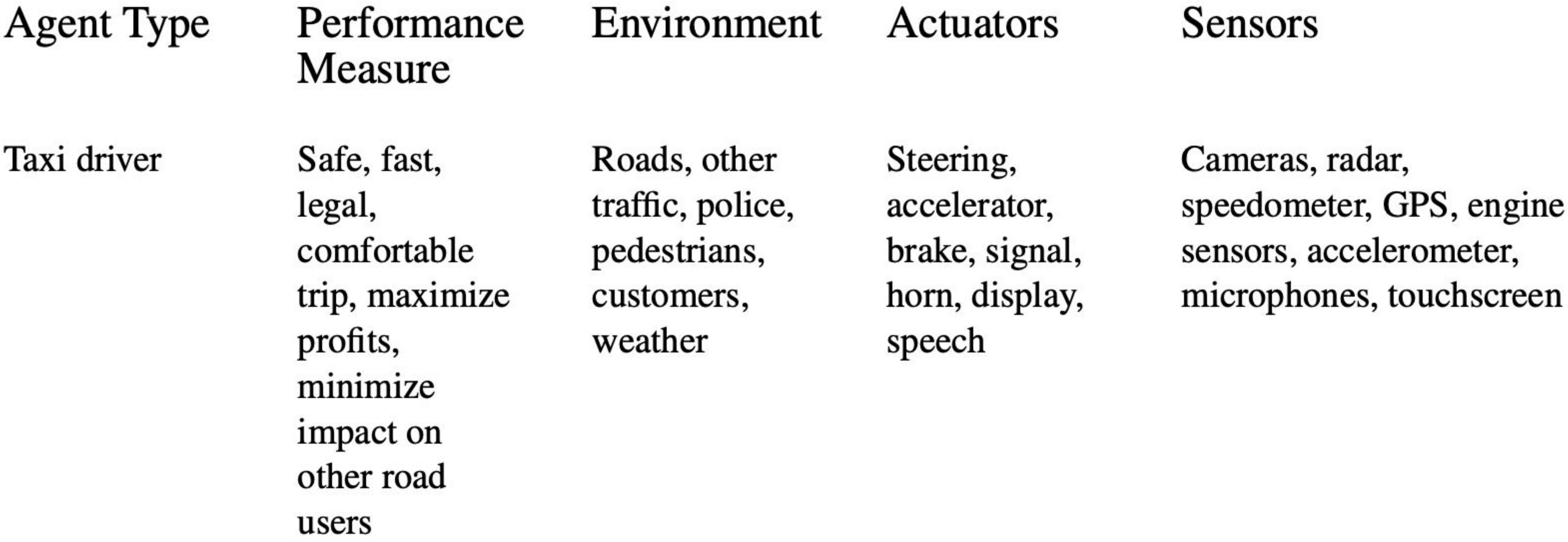
To design a rational agent,

we must specify the **task environment**

Task Environment

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

Example - An Automated Taxi Driver



Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits, minimize impact on other road users	Roads, other traffic, police, pedestrians, customers, weather	Steering, accelerator, brake, signal, horn, display, speech	Cameras, radar, speedometer, GPS, engine sensors, accelerometer, microphones, touchscreen

Source: Russell and Norvig, *AIMA, 2020*

Task Environment Types

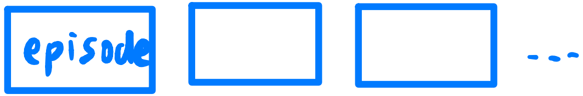
- Fully observable vs. partially observable
    - Fully observable
        - If an agents’ sensors give it access to the complete state of the environment at each point in time
    - Partially observable
        - e.g., a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares

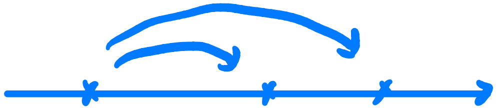
Task Environment Types (cont.)

- Deterministic vs. nondeterministic
    - Deterministic
        - The next state of the environment is completely determined by the current state and the action
    - Nondeterministic
        - e.g., there’s a chance of rain tomorrow
          (the possibilities are listed without being quantified)
        - **Stochastic**
            - A model of environment explicitly deals with probabilities

Task Environment Types (cont.)

- Episodic vs. sequential
  - Episodic
    - In each episode, the agent receives a percept and then performs a single action
    - The next episode does not depend on the actions taken in previous episodes
    - e.g., many classification tasks
  - Sequential
    - The current decision could affect all future decisions





Task Environment Types (cont.)

(a)

(v) engage in long & careful consideration

↑

- Static vs. dynamic
  - Dynamic
    - If the environment can change while an agent is deliberating, the environment is dynamic for that agent
  - **Semidynamic**
    - The environment does not change with time
    - The agent’s performance score changes with the time

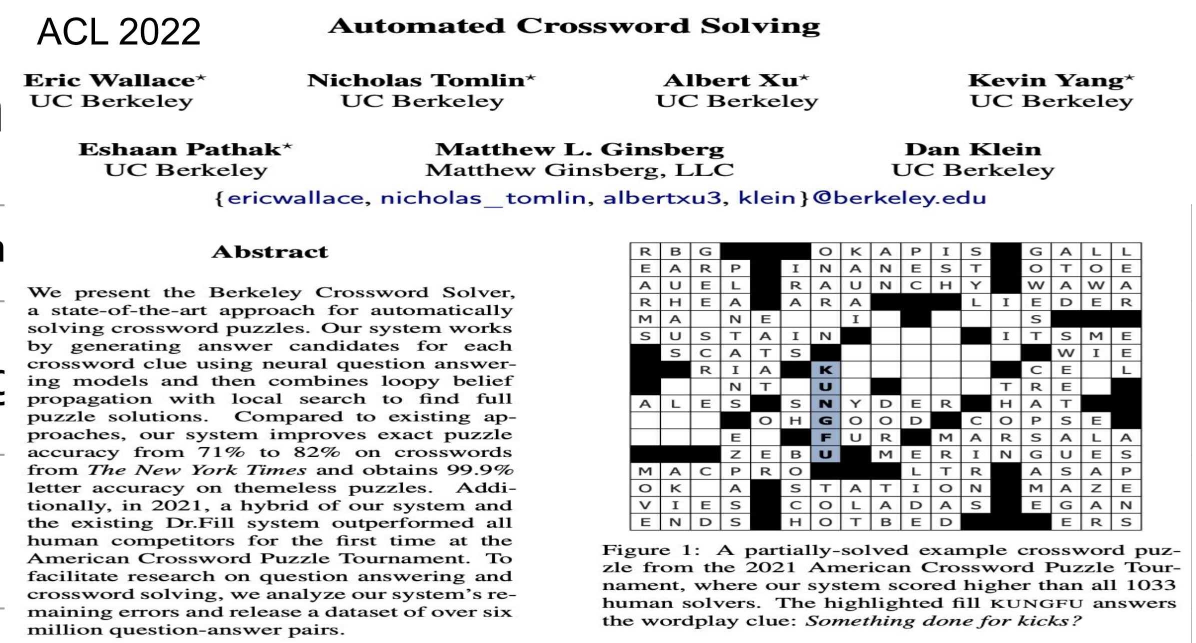
Task Environment Types (cont.)

- Discrete vs. continuous
    - Discrete
        - e.g., the chess environment has a finite number of distinct states
    - Continuous
        - e.g., taxi driving is a continuous-time problem
            - A continuous-state
            - Actions are continuous (steering angles, etc.)
- Single-agent vs. multiagent

Examples of Task Environment Types

	Observable?	Agents?	Deterministic?	Episodic?	Static?	Discrete?
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Go with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous

Examples of Task En



ACL 2022

Automated Crossword Solving

Eric Wallace* UC Berkeley
Nicholas Tomlin* UC Berkeley
Albert Xu* UC Berkeley
Kevin Yang* UC Berkeley
Eshaan Pathak* UC Berkeley
Matthew L. Ginsberg Matthew Ginsberg, LLC
Dan Klein UC Berkeley
{ericwallace, nicholas_tomlin, albertxu3, klein}@berkeley.edu

Abstract

We present the Berkeley Crossword Solver, a state-of-the-art approach for automatically solving crossword puzzles. Our system works by generating answer candidates for each crossword clue using neural question answering models and then combines loopy belief propagation with local search to find full puzzle solutions. Compared to existing approaches, our system improves exact puzzle accuracy from 71% to 82% on crosswords from *The New York Times* and obtains 99.9% letter accuracy on themeless puzzles. Additionally, in 2021, a hybrid of our system and the existing Dr.Fill system outperformed all human competitors for the first time at the American Crossword Puzzle Tournament. To facilitate research on question answering and crossword solving, we analyze our system's remaining errors and release a dataset of over six million question-answer pairs.

Figure 1: A partially-solved example crossword puzzle from the 2021 American Crossword Puzzle Tournament, where our system scored higher than all 1033 human solvers. The highlighted fill KUNGFU answers the wordplay clue: *Something done for kicks?*

	Observable?	Agen				
Crossword puzzle	Fully	Sing				
Go with a clock	Fully	Mu				
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous

Real world: partially, multi-agent, nondeterministic, sequential, dynamic, continuous

Agent Structure

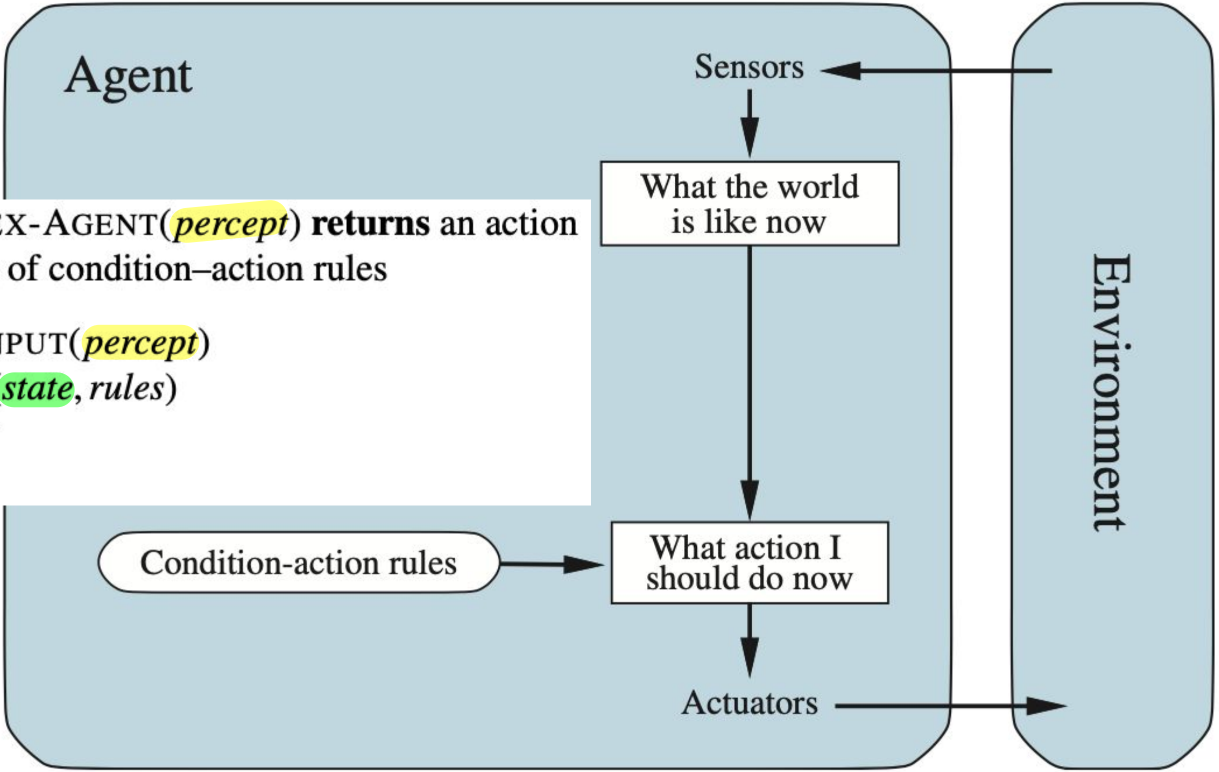
- Agent = architecture + program
    - Agent architecture includes physical sensors and actuators
    - **Agent program** runs on the physical architecture to implement the agent function (the job of AI)

Agent Types

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents

Credit: Dreamstime.com

Simple Reflex Agent

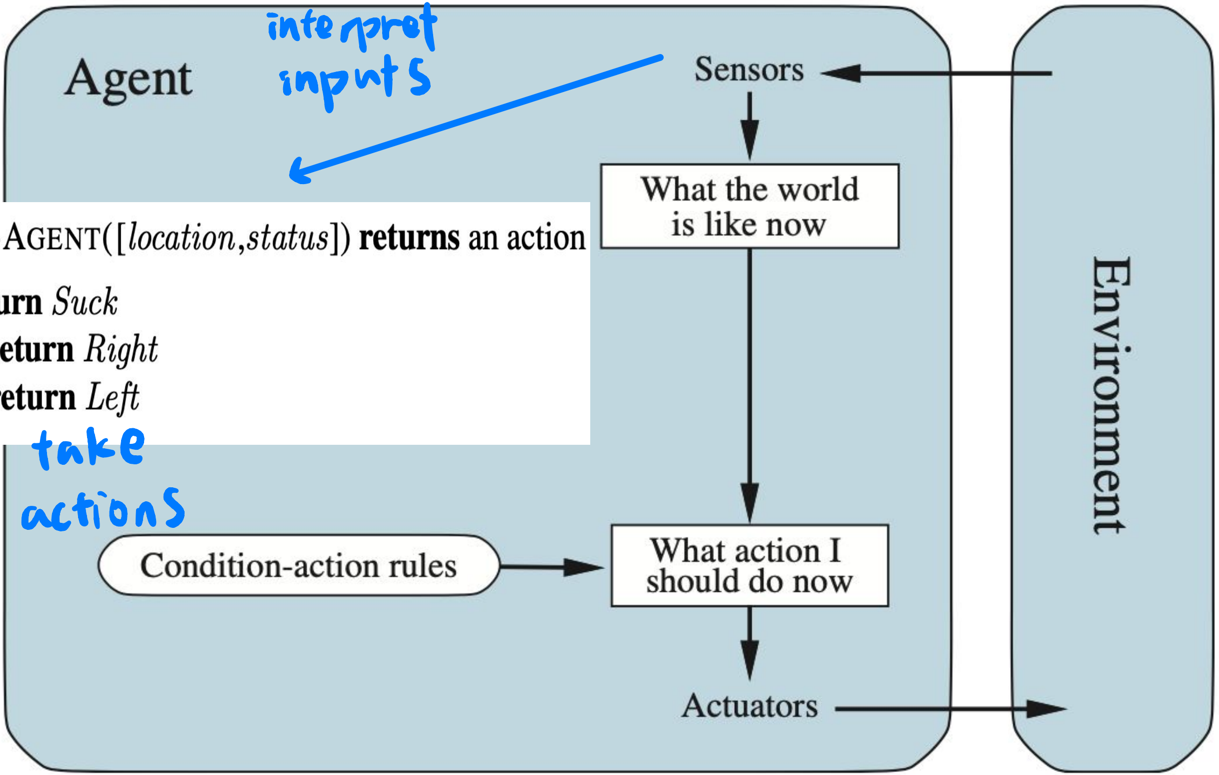


```
function SIMPLE-REFLEX-AGENT(percept) returns an action
  persistent: rules, a set of condition–action rules

  state ← INTERPRET-INPUT(percept)
  rule ← RULE-MATCH(state, rules)
  action ← rule.ACTION
  return action
```

Source: Russell and Norvig, *AIMA, 2020*

Simple Reflex Agent



interpret inputs

```
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
```

rule match

take actions

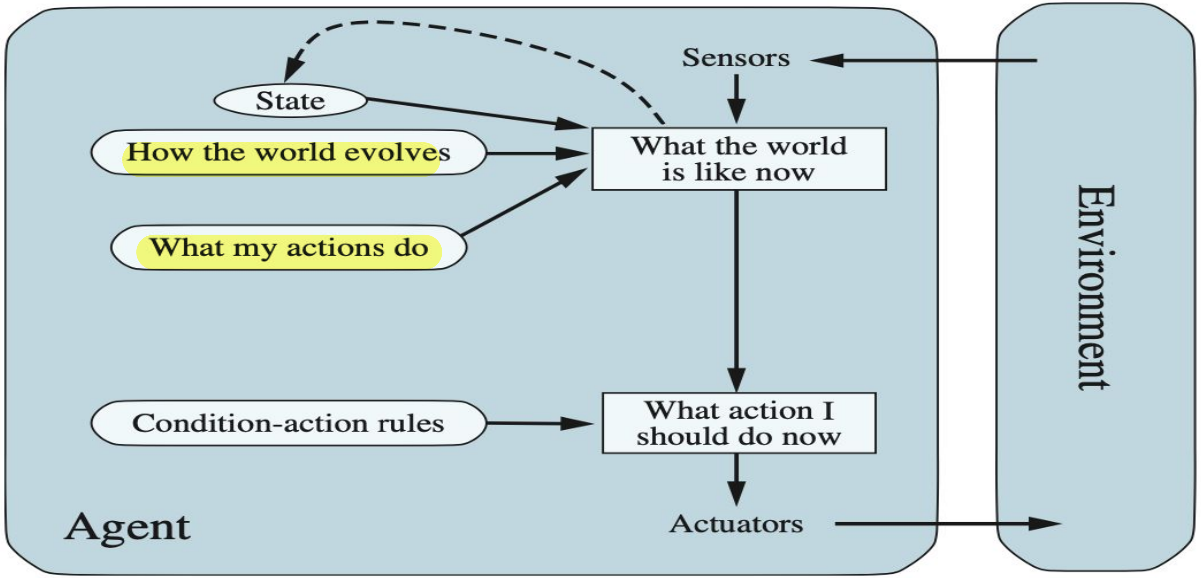
Source: Russell and Norvig, *AIMA, 2020*

Credit: Dreamstime.com

Model-based Reflex Agent

- Keep track the part of the world it cannot see now using an **internal model** (Partial observability)
- Then choose an action in the same way as the reflex agent

Knowledge:
Transition model of the world
Sensor model



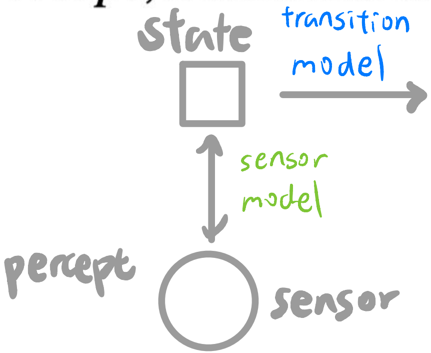
Source: Russell and Norvig, *AIMA, 2020*

```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
    persistent: state, the agent's current conception of the world state
                transition_model, a description of how the next state depends on
                        the current state and action
                sensor_model, a description of how the current world state is reflected
                        in the agent's percepts
                rules, a set of condition–action rules
                action, the most recent action, initially none

    state ← UPDATE-STATE(state, action, percept, transition_model, sensor_model)
    rule ← RULE-MATCH(state, rules)
    action ← rule.ACTION
    return action
```

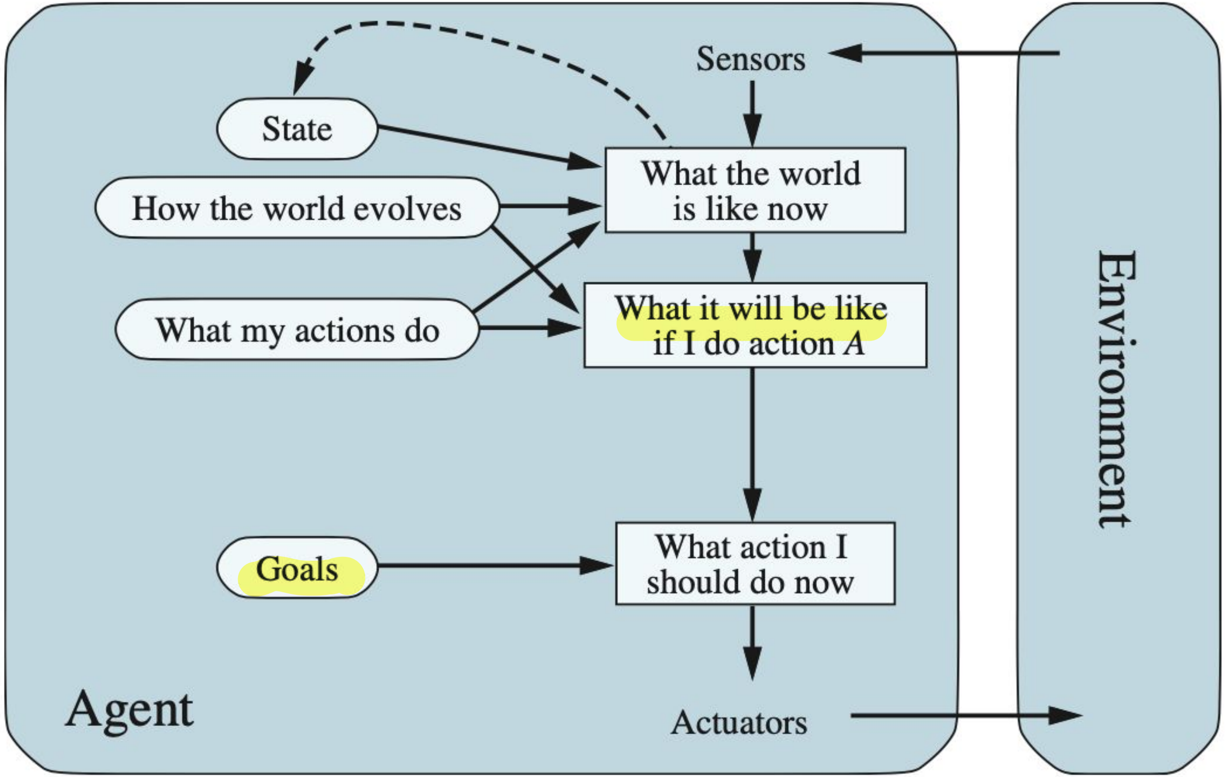
internal model

choose an action the same way as an reflex agent



Credit: Dreamstime.com

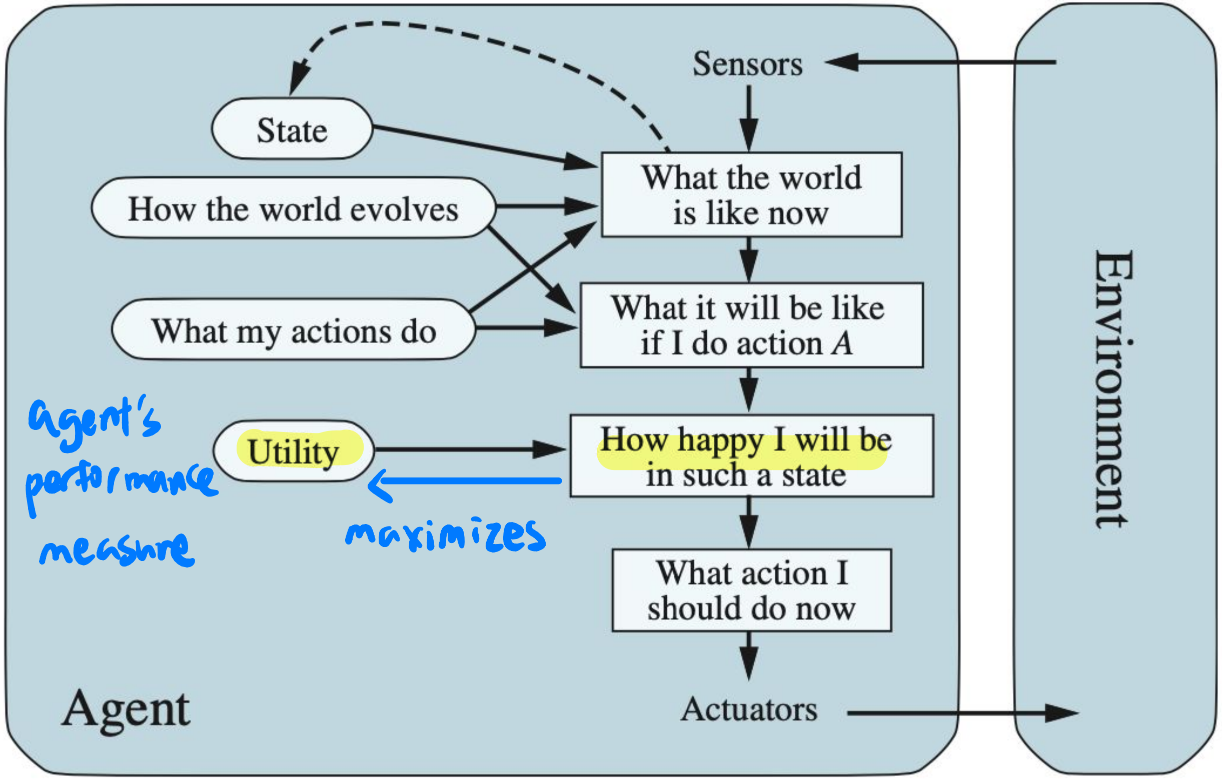
Goal-based Agent


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

Source: Russell and Norvig, *AIMA, 2020*

Credit: Dreamstime.com

Utility-based Agent

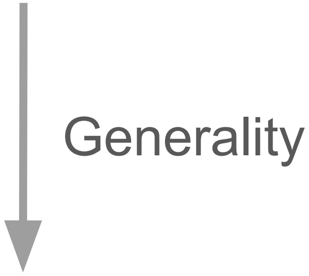

Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action *A*
Utility
How happy I will be in such a state
What action I should do now
Actuators
Agent
Environment
agent's performance measure
maximizes

Source: Russell and Norvig, *AIMA*, 2020

Agent Types

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents

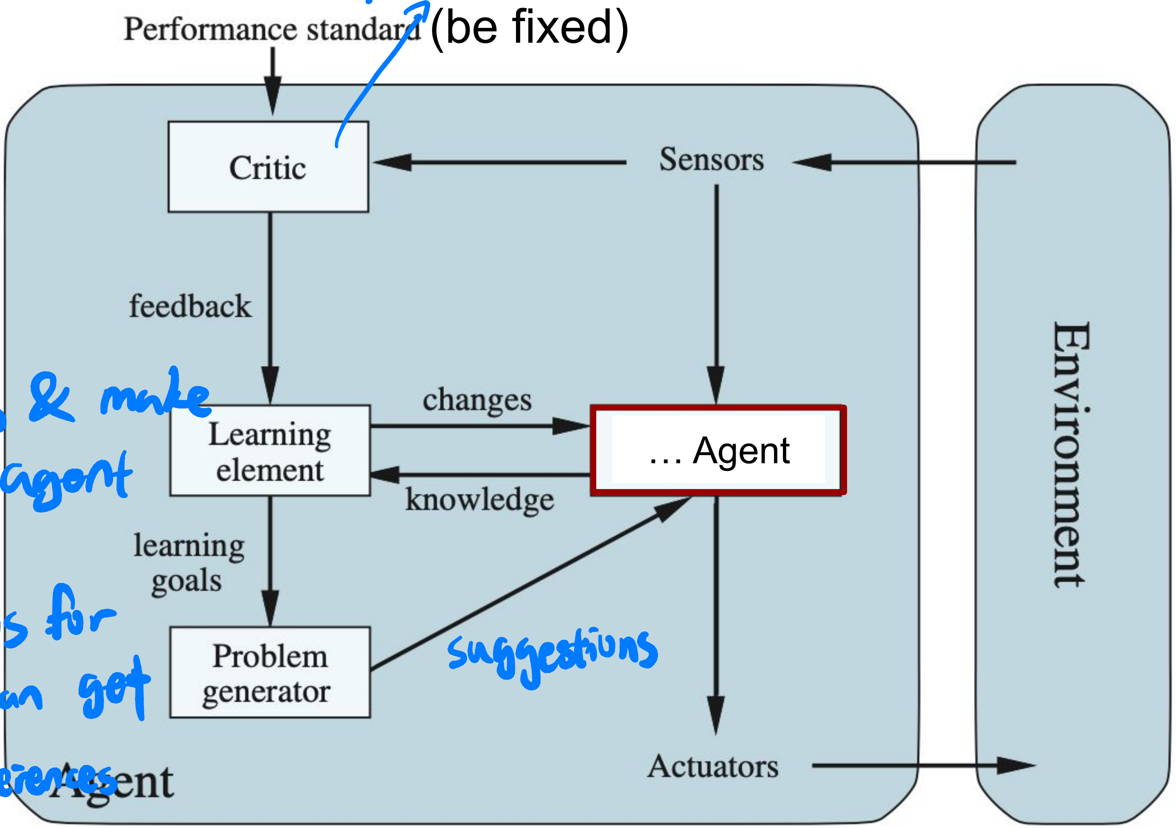
⇒ Learning agents



Credit: Dreamstime.com

Learning Agent

Finds out how well the agent is performing & provides feedback based on fixed standards



receive feedbacks & make changes to the agent

Suggesting actions for the agent so it can get some fresh experiences

Source: Russell and Norvig, *AIMA*, 2020