CS 461
ARTIFICIAL INTELLIGENCE

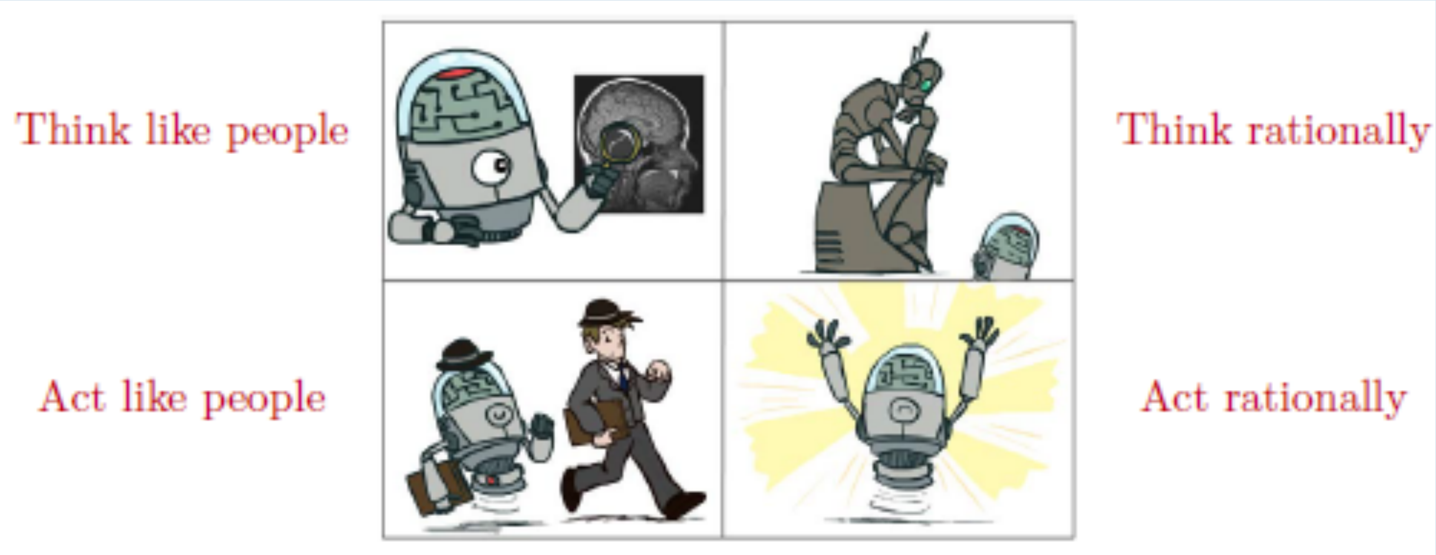
Lecture # 02
March 04, 2021
SPRING 2021
FAST – NUCES, CFD Campus

Dr. Rabia Maqsood
rabia.maqsood@nu.edu.pk

Today’s Topics

- Quick recap: Approaches to AI & The Turing Test
- History of AI
- Rational agents
- Task Environments
- Nature of environments
  - *PEAS*
  - *Properties*
- Rational agent types

Approaches to AI

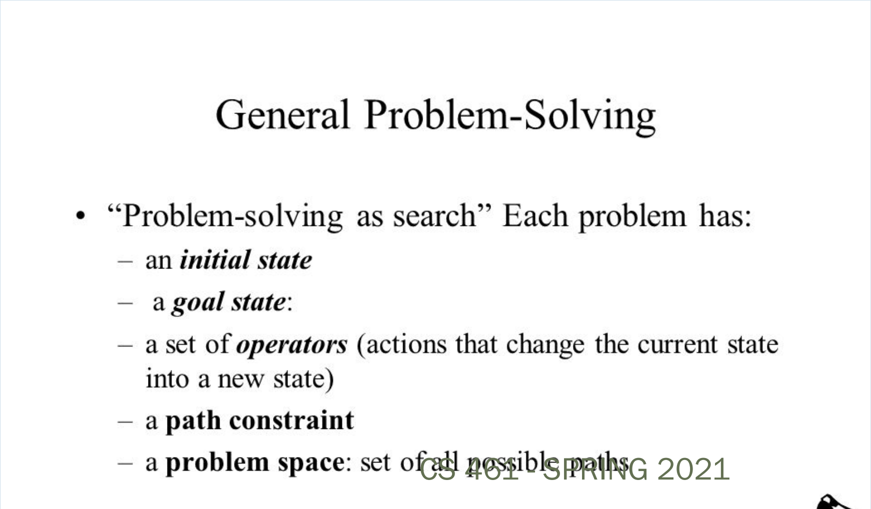


“[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning…” (Bellman, 1978)	“The study of mental faculties through the use of computational models” (Charniak & McDernott, 1985)
”The study of how to make computers do things at which , at the moment, people are better” (Rich & Knight, 1991)	“The branch of computer science that is concerned with the automation of intelligent behavior” (Luger & Stubblefield, 1993)

HISTORY OF AI

Abridged history of AI

- 1943 McCulloch & Pitts: Boolean circuit model of brain
- 1950 Turing's "Computing Machinery and Intelligence"
- 1956 Birth of AI – Dartmouth: "Artificial Intelligence" adopted
- 1952-69 Look, Ma, no hands! (early enthusiasm, great expectations)
- 1950s Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist followed by General Problem Solver (the "first AI program"), McCarthy introduced LISP language, Slagle's SAINT program, and many more.


General Problem-Solving

- "Problem-solving as search" Each problem has:
  - an ***initial state***
  - a ***goal state***:
  - a set of ***operators*** (actions that change the current state into a new state)
  - a **path constraint**
  - a **problem space**: set of all possible paths

Abridged history of AI

- 1966-73 AI discovers computational complexity
  No progress seemed in *Machine Evolution*
  Neural network research almost disappears
- 1969-79 Early development of knowledge-based systems
- 1980-- AI becomes an industry,
  Soon after that, “AI Winter” started
- 1987-- AI becomes a science
- 1995-- Emergence of intelligent agents
- 2001-- Availability of very large data sets

What is Intelligence?

- **Recall this definition**

Intelligence is a measure of the success of an entity in achieving its objectives by interacting with its environment.

Reveals important points…

- Presence of an environment to observe intelligent behavior
- Measure goals on a scale to measure intelligence

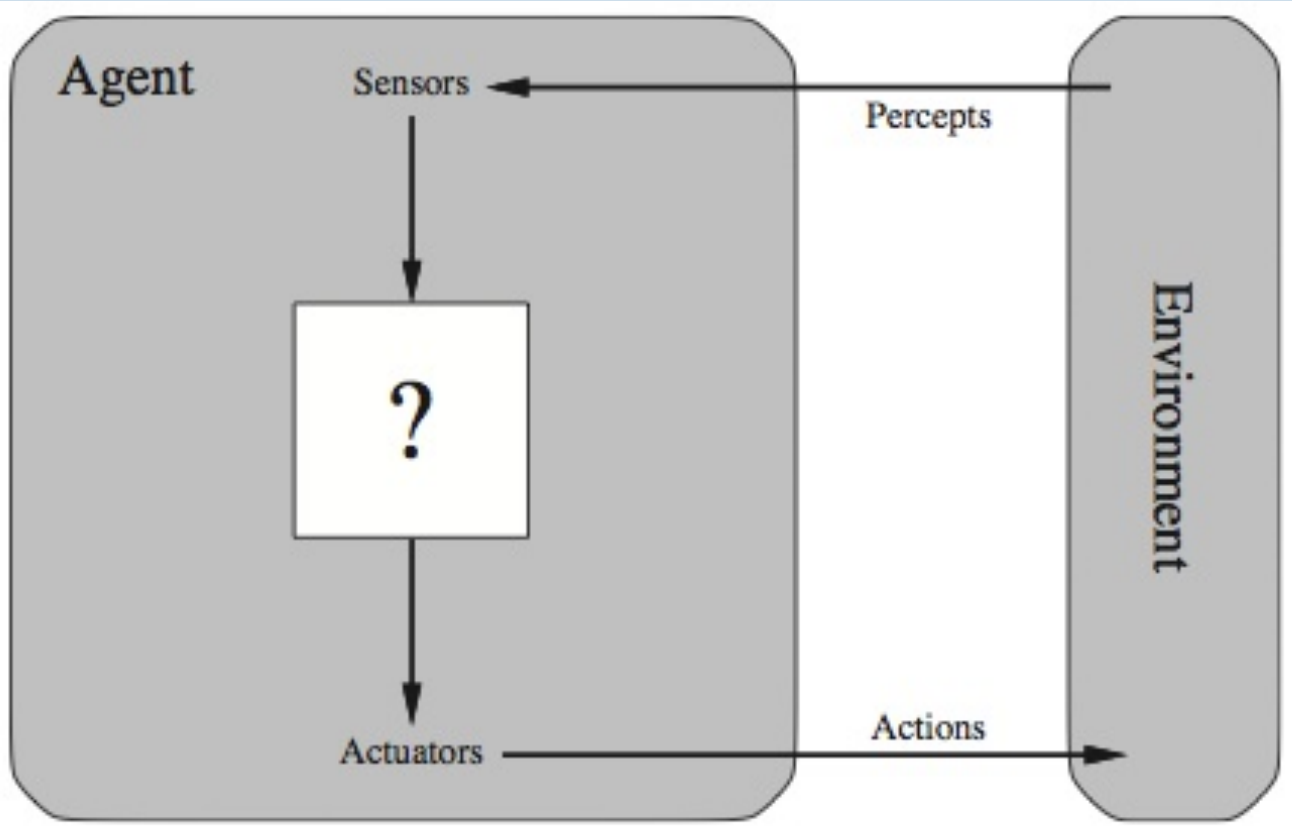
RATIONAL AGENT & ENVIRONMENTS

Agents and Environments

- An **agent** is an entity that perceives its ***environment*** through *sensors* and acts upon it through *actuators*.

Agents and Environments

Think of a human, robotic or software agent



Source: S. Russell & P. Norvig

An agent together with its environment is called a **world**.

Few terminologies

- Percept: agent's perceptual input at any given stage
- Percept sequence: complete history of everything the agent has ever perceived
- Agent function: describes agent's behavior at an abstract level
- Agent program: concrete implementation of an agent function

Agent Function & Agent Program

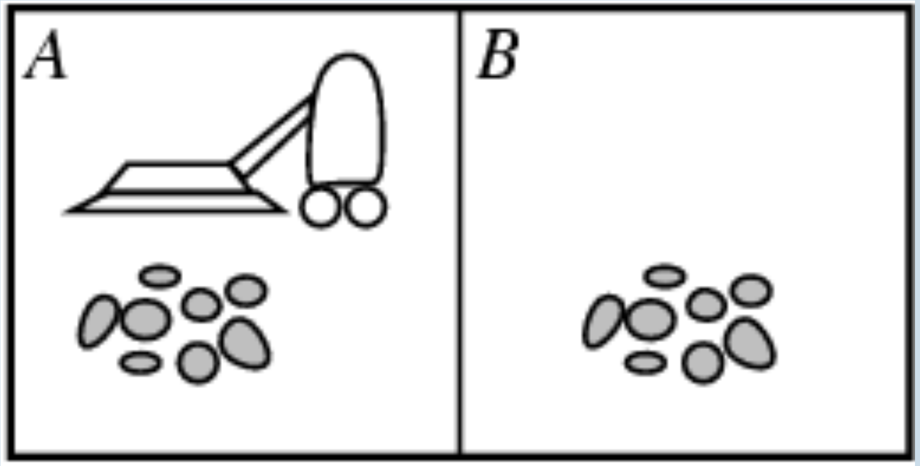
- Abstractly, an agent is a function from percept histories to actions:

  $$f : P^* \rightarrow A$$

  - *Usually represented by tabulation*

- Agent program is a concrete implementation of agent function

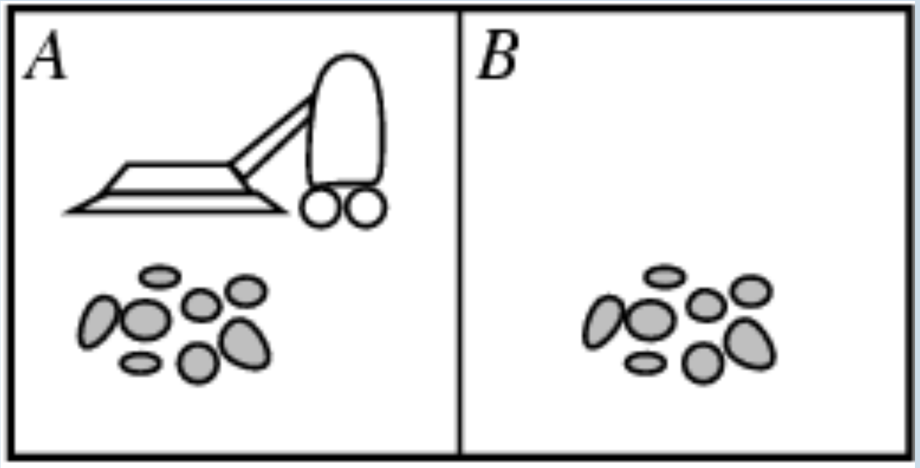
Vacuum Cleaner World Toy-Example



Two locations:	A and B
Percepts:	location and contents (dirt/clean)
Actions:	Left, Right, Suck
Function:	“Suck” if current square is dirty, “move” to the other square otherwise

Source: S. Russell & P. Norvig

Vacuum Cleaner World Toy-Example



Possible Precept Sequences	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[A, Clean], [B, Dirty]	Suck
…	…

Source: S. Russell & P. Norvig

Rational Agent

- An agent which does the right thing.
- Performance measure evaluates any given sequence of environment states.
- Rational agent acts to **maximize** the expected value of performance measure.

Rationality

- Rationality at any given time depends on four things:
  1. *Performance measure that defines the criterion of success*
  2. *Agent's prior knowledge of the environment*
  3. *Actions that the agent can perform*
  4. *Agent's percept sequence to the date*

Few terminologies

- Information gathering/Exploration: doing actions in order to modify future percepts
- Omniscience: knowing the actual outcome of action(s) and acting accordingly
  - *But impossible in reality*
- Learning: not restricting the agent for information gathering only but to make it learn as much as possible from what it perceives
- Autonomy: ability to learn & adapt something *(based on its own experience)* other than prior knowledge

Understanding Rational

- Rational ≠ omniscient
  - *Impossible in reality, rationality is limited by the available percepts*

- Rational ≠ perfection
  - *Focus is on maximizing expected performance measure rather than actual performance*

- Rational → exploration, learning, autonomy

Rational Agent

- For each possible percept sequence, a **rational agent** should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

Task Environments

- These are essentially the “problems” to which rational agents are the “solutions”.

Task Environments

PEAS

- Performance
- Environment
- Actuators
- Sensors

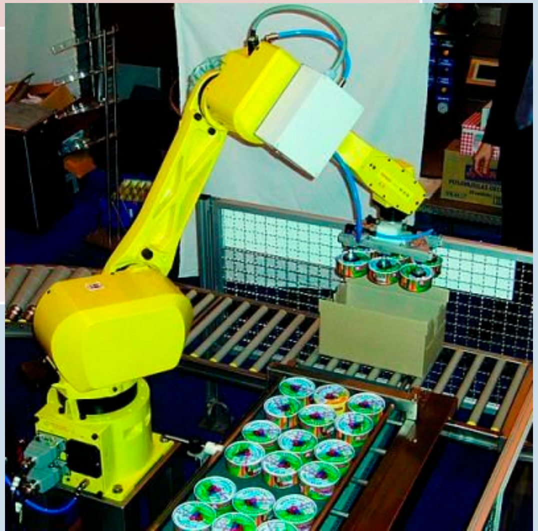
PEAS for taxi driver

Agent Type: Taxi Driver	
Performance measure	Safe, fast, legal, comfortable trip
Environment	Roads, other traffic, pedestrians, customers
Actuators	Steering, accelerator, brake, signal, horn, display
Sensors	Cameras, sonar, speedometer, GPS, odometer, engine sensors, etc.

http://static8.depositphotos.com/1264395/895/v/950/depositphotos_8953725-Taxi-Driver.jpg

PEAS for part-picking robot

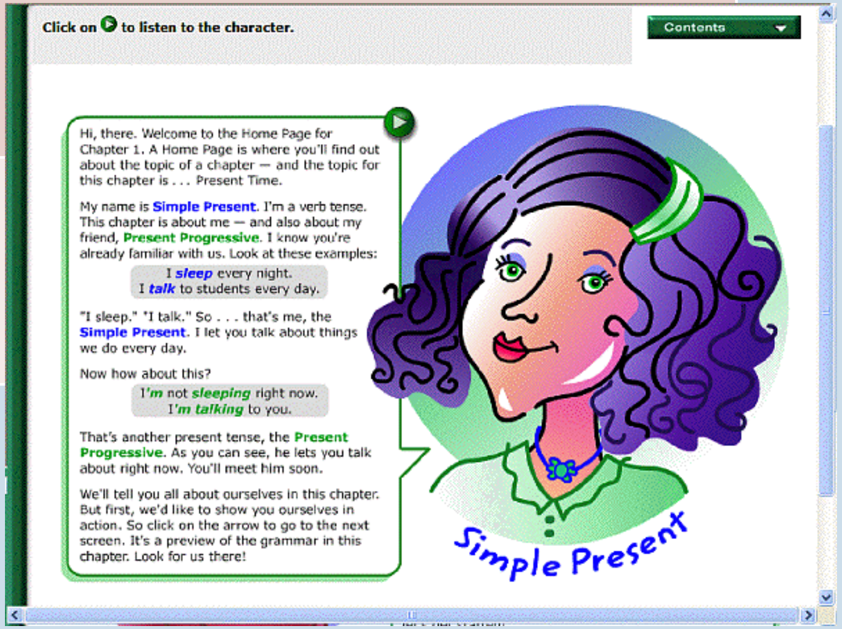
Agent Type: Part-picking robot	
Performance measure	Percentage of parts in correct bins
Environment	Conveyor belt with parts, bins
Actuators	Jointed arm and hand
Sensors	Camera, joint angle sensors



http://www.robots.com/images/FANUC%20Pick%20and%20Place%20Robot.jpg

PEAS for interactive English tutor

Agent Type: Interactive English tutor	
Performance measure	Maximize student's score on test
Environment	Group of students
Actuators	Screen display
Sensors	Keyboard



http://tesl-ej.org/ej42/m4pix/m4fig2.gif

Properties of task environment

- Fully observable vs. Partially observable
- Single vs. Multi-agent
- Deterministic vs. Stochastic
- Episodic vs. Sequential
- Static vs. Dynamic vs. Semi-dynamic
- Discrete vs. Continuous

Environment Types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable			
Deterministic			
Episodic			
Dynamic			
Discrete			
Single agent			

An environment is full observable when the sensors can detect all aspects that are relevant to the choice of action.

Environment Types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic			
Episodic			
Dynamic			
Discrete			
Single agent			

An environment is deterministic if the next environment state is completely determined by the current state and the action to be executed.

Environment Types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic			
Dynamic			
Discrete			
Single agent			

In an episodic environment the agent's experience can be divided into atomic steps where the agent perceives and perform an action. The choice of action depends only on episode itself.

Environment Types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Dynamic			
Discrete			
Single agent			

A dynamic environment can change while the agent is choosing its action. It is semi-dynamic if the agent's performance changes even if the environment remains the same.

Environment Types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Dynamic	Semi	No	Yes
Discrete			
Single agent			

*Discrete/continuous distinction applies to the **state** of the environment to the way **time** is handled, and to the **percepts** and **actions** of the agent.*

Environment Types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Dynamic	Semi	No	Yes
Discrete	Yes	Yes	No
Single agent			

Does the environment contain other agents who are also maximizing some performance measure that depends on the current agent's actions?

Environment Types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Dynamic	Semi	No	Yes
Single agent	No	No	No

The environment type largely determines the agent design.

Environment Types

- The simplest environment is:

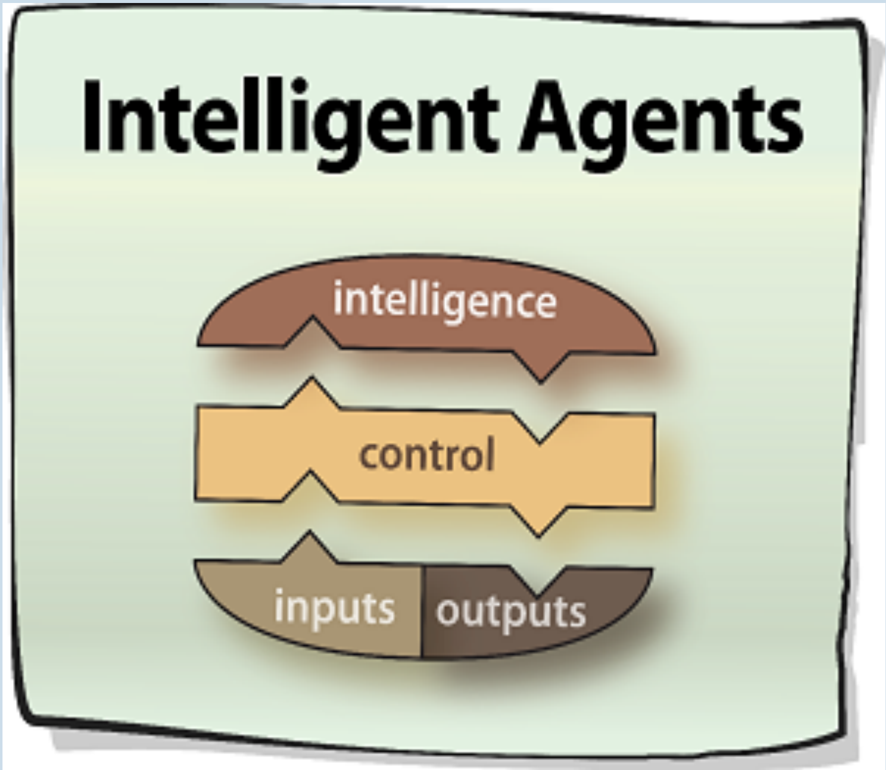
Fully observable, deterministic, episodic, static, discrete and single-agent

- Most real situations are:

Partially observable, stochastic, sequential, dynamic, continuous and multi-agent

The environment type largely determines the agent design.

RATIONAL AGENTS TYPES



http://research.microsoft.com/en-us/projects/marlowe/intelligentagentssmall.png

Agent = Architecture + Program

Table-lookup Agent

- Simplest possible agent function:
  - *All possible states and their optimal actions specified by the designers in advance*

```
function TABLE-DRIVEN-AGENT(percept) returns an action
  persistent: percepts, a sequence, initially empty
              table, a table of actions, indexed by percept sequences, initially fully specified

  append percept to the end of percepts
  action ← LOOKUP(percepts, table)
  return action
```

This approach is doomed to failure

Table-lookup Agent

- Drawbacks:
  - *Huge table (consider continuous states)*
  - *Could take a long time to build the table*
  - *No autonomy!*
  - *Even with learning, agent could need a long time to learn the table entries*

We rather need more sophisticated agent models.

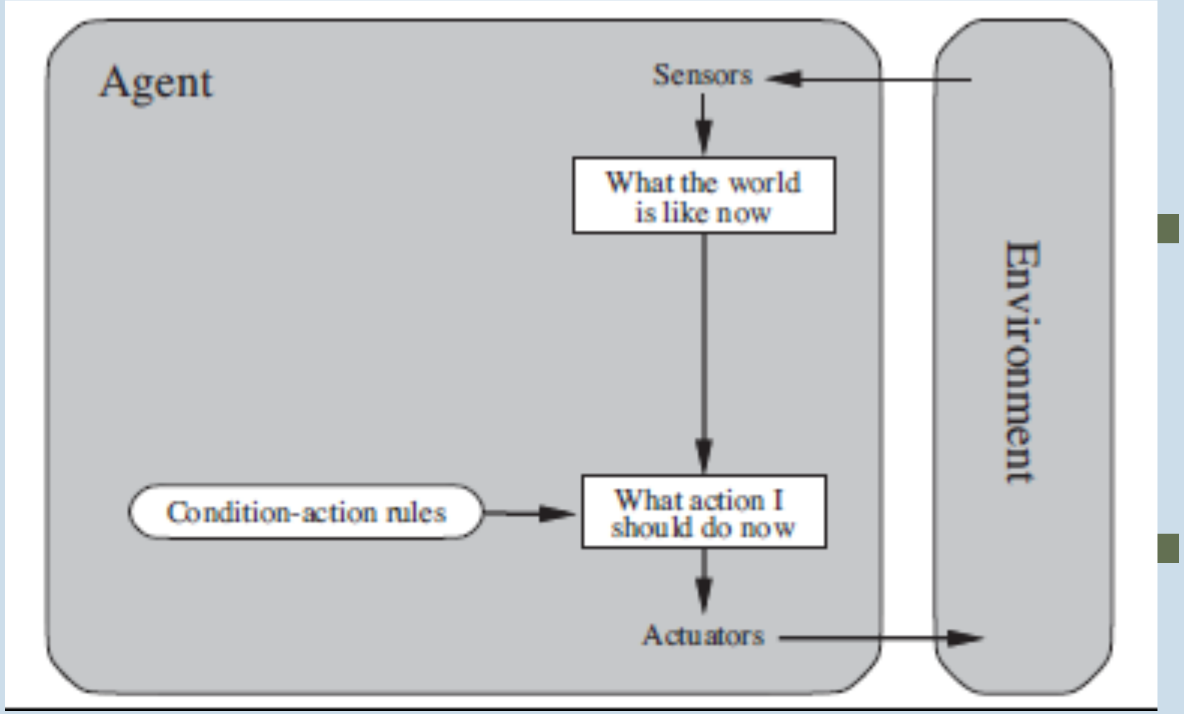
Agent types

- Four basic kind of agent programs will be discussed:
  - *Simple reflex agents*
  - *Model-based reflex agents*
  - *Goal-based agents*
  - *Utility-based agents*

- All these can be turned into learning agents.

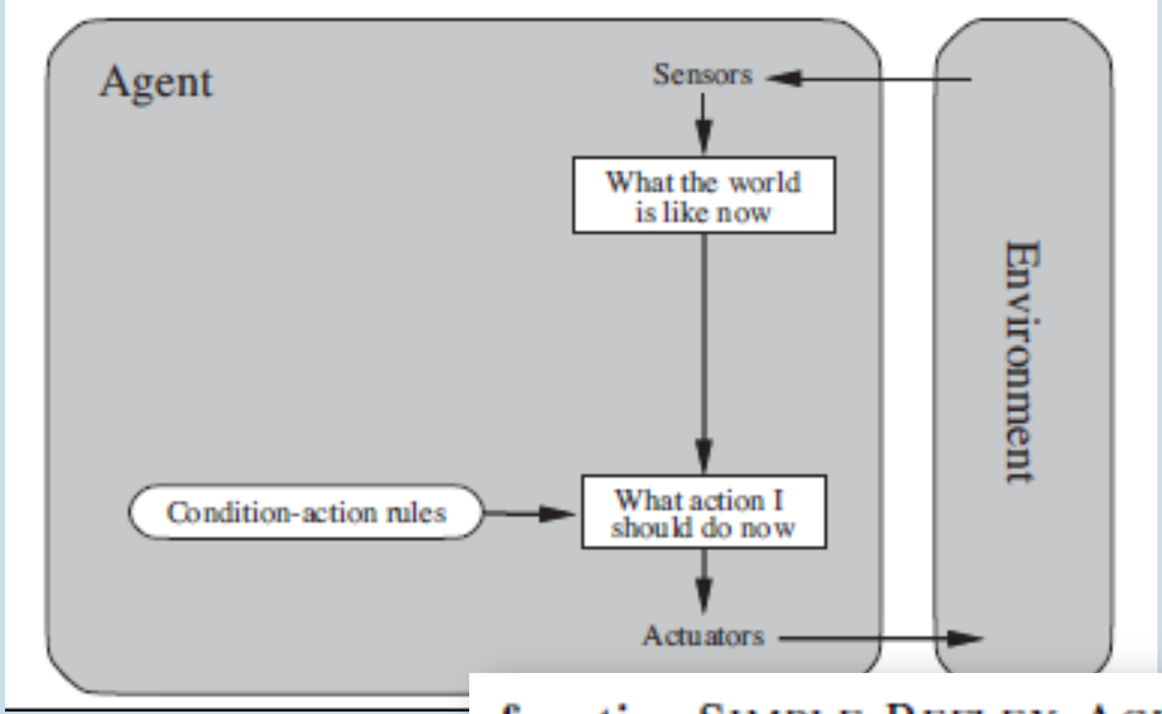
Simple reflex agent

Source: S. Russell & P. Norvig



- Select action is based *only on the current* percept
- Large reduction in possible percept/actions situations
- Implemented through *condition-action rules e.g. if dirty then suck, etc.*

Simple reflex agent



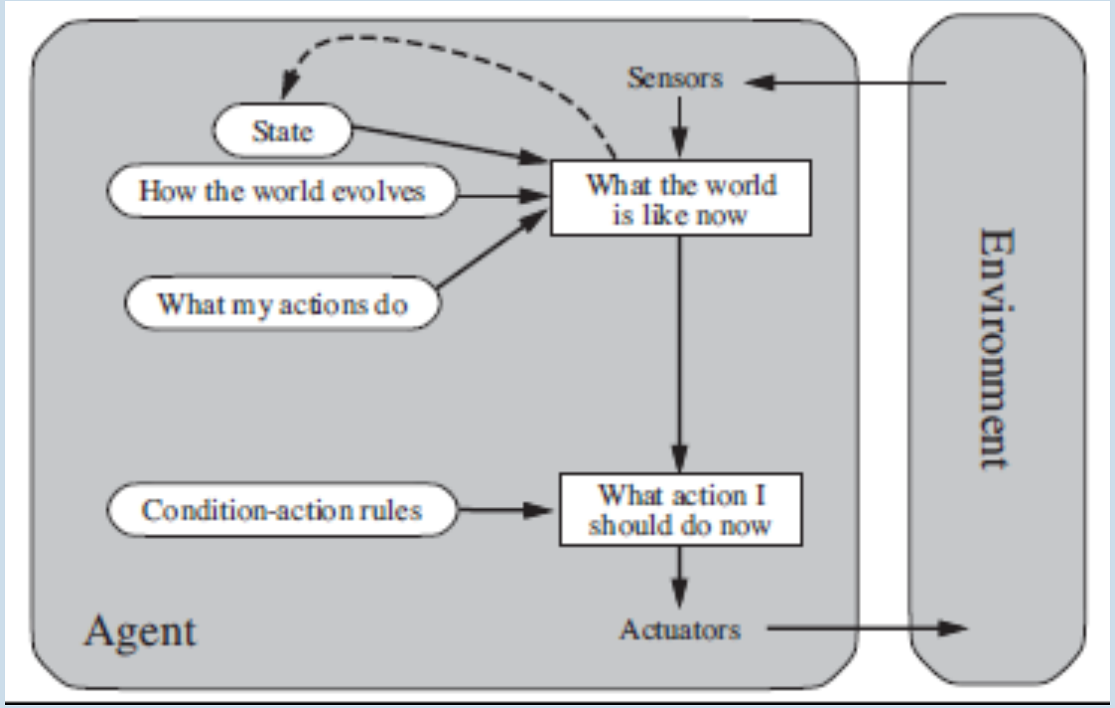
function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *rules*, a set of condition–action rules

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

Source: S. Russell & P. Norvig

Model-based reflex agent

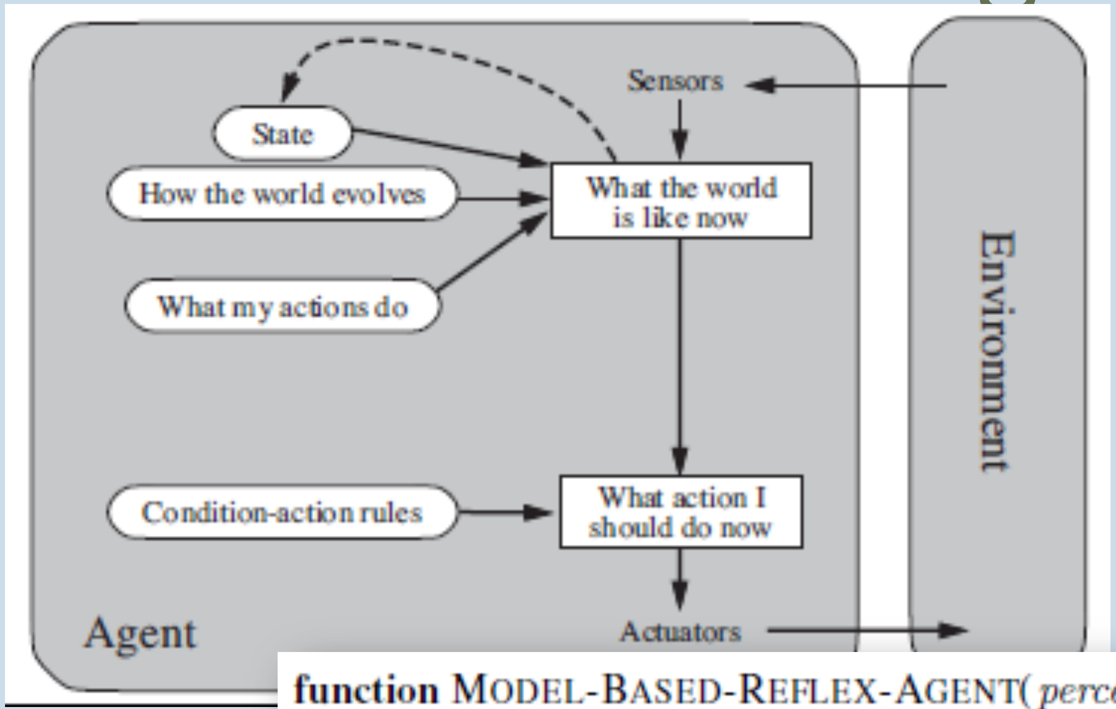
Reflex + State



Source: S. Russell & P. Norvig

- To tackle *partially observable* environments (by maintaining internal state)
- Over time update state using world knowledge
  - *How the world evolves*
  - *How do my actions affect world*

Model-based reflex agent

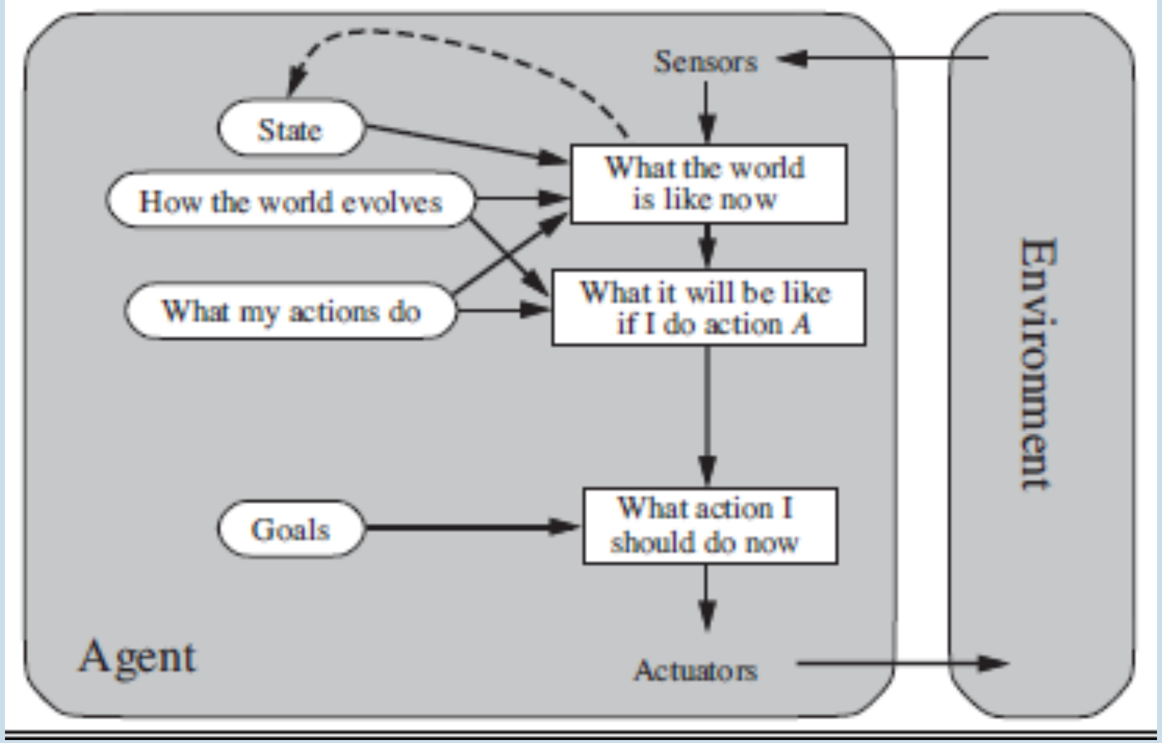



```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
              model, a description of how the next state depends on current state and action
              rules, a set of condition–action rules
              action,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

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

Source: S. Russell & P. Norvig

Goal-based agent

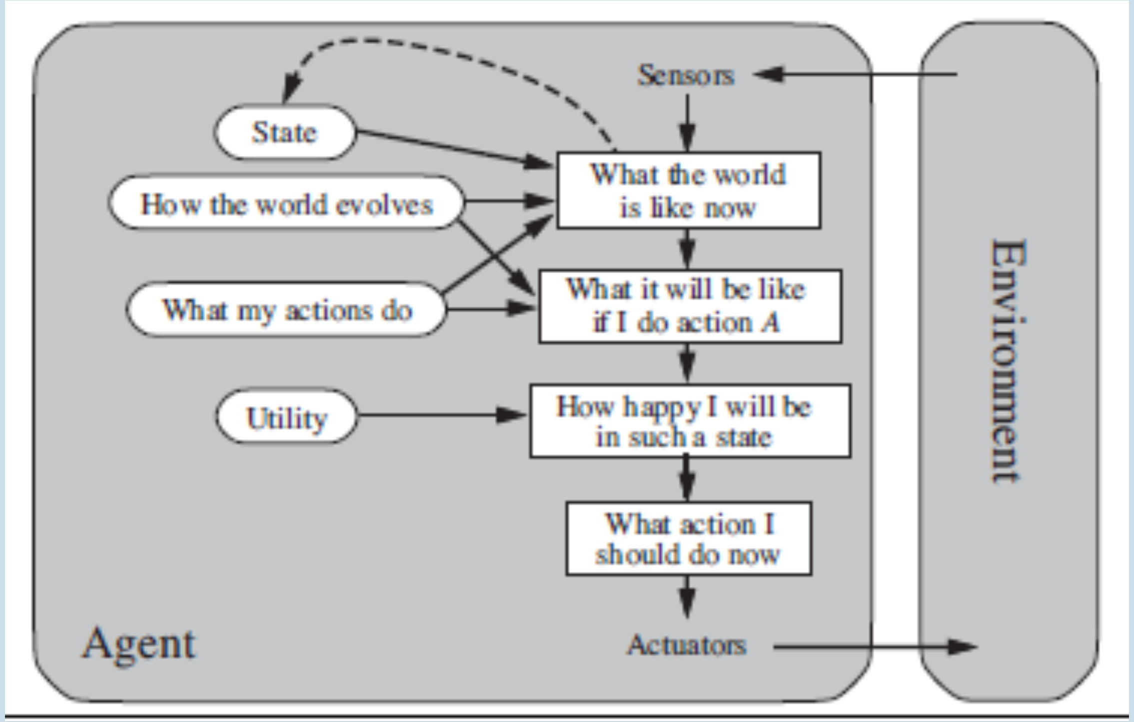


Source: S. Russell & P. Norvig

- The agent needs a goal to know which states are *desirable*
- Typically investigated in Search and Planning subfields
- More flexible since knowledge is represented explicitly and can be manipulated
- Can easily be changed for new goals as compared to simple reflex agent (which requires new rules to be written)

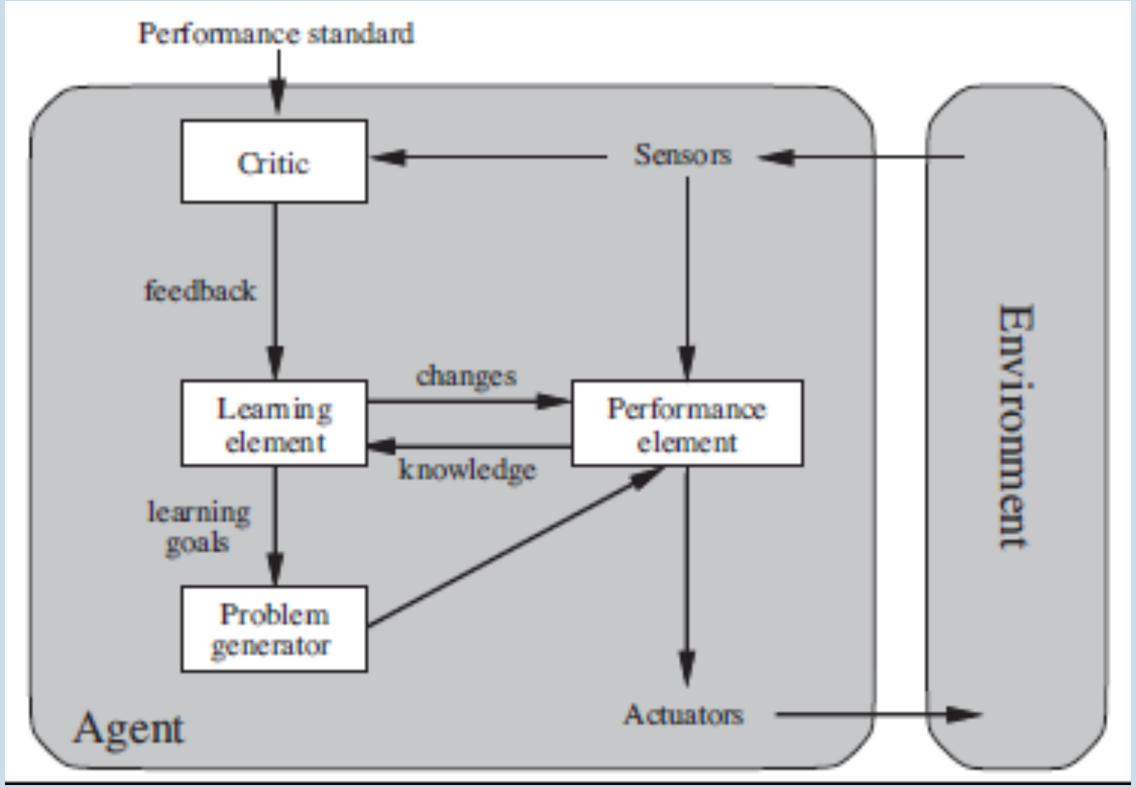
Utility-based agent

Source: S. Russell & P. Norvig



- Certain goals can be achieved in different ways
- Utility function maps a (sequence of) state(s) onto a real number
- Improves on goals:
  - *Selecting between conflicting goals*
  - *Select appropriately between several goals based on likelihood of success*

Learning agent



Source: S. Russell & P. Norvig

- Learning element: responsible for making improvements
- Performance element: selects external actions based on percepts
- Critic: determines how the agent is doing and determines how it should modify performance element to do better in the future
- Problem generator: suggests actions that will lead to new and informative experience

Examples

- Simple reflex agents
  - *If temperature > 25, turn on the air conditioner*
- Model-based reflex agents
  - *Turn on the air conditioner at 9 a.m.*
- Goal based agents
  - *Keep people in the room comfortable*
- Utility based agents
  - *Measure comfortable with a comfortable utility function (comfortable or uncomfortable is not enough)*
- Learning agents
  - *Learn the definition of "comfortable" with some feedback*

Tentative Marks Distribution

Item Name	Quantity	Marks (%)
Home work	Very often	0%
Quizzes	05-06 [all announced]	10%
Assignments	05	15%
Mid Exams	02	30% [15 abs.+15 abs.]
Final Exam	01	45%

No retakes for missed quizzes
Assignment late work policy: 30% marks deduction (per day after the deadline)

- Enjoy the following TED talk!

http://www.ted.com/talks/jeremy_howard_the_wonderful_and_terrifying_implications_of_computers_that_can_learn

Reading Material

- Russell & Norvig: Chapter # 1, 2
- David Poole: Chapter # 1