# IS/SE/SS ZC444: Artificial Intelligence

# 03 Types of AI Agents

**Dr. Kamlesh Tiwari**
Associate Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

August 14, 2023 ONLINE WILP @ BITS-Pilani [July-Dec 2023]

http://ktiwari.in/ai

## Intelligent Agents

In pursuit of computers doing things which at the moment, people do better, AI attempts to build intelligent entities called **Agents** [1]

Agent perceives the environment through **sensors** and act upon the environment through **actuators**

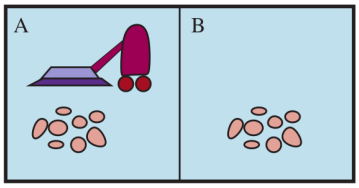
- Our approach is to build **rational agent**
- How well agent can behave depends on the nature of environment. Some environments are more difficult.

Agents choice of action can depend on percept sequence.

[1] Consider human, robot or software agent



## Example: Toy Vacuum Cleaner




Two cells, dirt/or-not. Can sense dirt and move

- If current sequence is dirty, then suck; otherwise move to other square.
- What makes this agent intelligent?



## Tabulation of Vacuum Cleaner World

| Percept | Action |
|---|---|
| [A,Clean] | Right |
| [A,Dirty] | Suck |
| [B,Clean] | Left |
| [B,Dirty] | Suck |
| ⋮ | |
| [A,Clean], [A,Clean] | Right |
| [A,Clean], [A,Dirty] | Suck |
| ⋮ | |

- Different agents could be defined by filling in the right-hand side column of the table in various way. ($actions^{historysize \times perceptStates}$)
- Question remains which agent is better



## Rational Agent

Rational Agent is one that does the right things

- Sequence of actions of agent leads to sequence of *states of the environment*
- A **performance measure** could be used to evaluate how the sequence of state of environment is desirable (NOT of agent)
- Design performance measure according to what one actually wants in the environment, rather then how agent should behave
- What is desired is not easy to define (simple life or ups and down) (every one in moderate poverty or some rich some more poor)

**Rationality depends on**

1. Performance measure that defines the criteria of success
2. Agents prior knowledge of the environment
3. The action that the agent can perform
4. Agents percept sequence till date



## 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.*

**Two performance measure for our Vacuum Cleaner agent**

- Rational if: awards one point for clean square at each time stamp
- Not rational if: awards one point for clean square at each time stamp and puts penalty for movement to left or right

Rationality is not **perfection**. Rationality maximizes expected performance whereas perfection maximizes actual performance.

## Omniscience, Learning and Autonomy

- Omniscient agent knows actual outcome of its actions (this is impossible)
- Information gathering is an important part of rationality (agent should get appropriate percept before taking action)
- Agent initial configuration could reflect some prior *knowledge*. The agent can modify it and augment
- In some cases knowledge about environment states could be there priory
- An **autonomous agent** learns to compensate for partial or incorrect prior knowledge

Task environment is the "problem" which the rational agent have to "solve"



## PEAS: Task Environment

**P**erformance, **E**nvironment, **A**ctuators, **S**ensors (PEAS) formally describes the task environment

### Consider Autonomous Taxi

- **Performance Measure**: safe, fast, legal, confortable, maximize profit
- **Environment**: roads, traffic, pedestrians, customers
- **Actuators**: Steering, accelerator, brake, signal, horn, display
- **Sensors**: Camera, sonar, speedometer, GPS, odometer, engine sensors, keyboard

Software agents (softbots) exists in rich and unlimited domain



## PEAS: Examples

| Agent Type | Performance Measure | Environment | Actuators | Sensors |
|---|---|---|---|---|
| Medical Diagnostic System | Healthy patient, Reduced cost | Patient, Hospital, Staff | Display of questions, tests, diagnoses, treatments, referrals | Keyboard entry of symptoms, findings, patient's answer |
| Satellite image analysis system | Correct image categorization | Downlink from orbiting satellite | Display of scene categorization | Color pixel array |
| Part-picking robot | percentage of parts in correct bins | Conveyor belt with parts, bins | Joined arm and hands | Camera, joint angle sensors |
| Refinery controller | Purity, yield, safety | Refinery, operators | Values, pumps, heaters, displays | Temperature, pressure, chemical sensors |
| Interactive English tutor | Student's score on test | Set of students, testing agency | Display of exercises, suggestions, corrections | Keyboard entry |



## Task Environment Properties

- **Fully Observable** vs **Partially Observable**
- **Single agent** vs **Multi agent**: competition, cooperation, communication
- **Deterministic** vs **Stochastic**: next state is uncertain, non-deterministic
- **Episodic** vs **Sequential**: uncertain, non-deterministic
- **Static** vs **Dynamic**
- **Discrete** vs **Continuous**
- **Known** vs **Unknown**



## Task Environment: Examples

| Task Environment | Observable | Agents | Deterministic | Episodic | Static | Discrete |
|---|---|---|---|---|---|---|
| Crossword puzzle | Fully | Single | Deterministic | Sequential | Static | Discrete |
| Chess with a clock | Fully | Multi | Deterministic | Sequential | Semi | Discrete |
| Taxi driving | Partially | Multi | Stochastic | Sequential | Dynamic | Continuous |
| Medical diagnostic | Partially | Single | Stochastic | Sequential | Dynamic | Continuous |
| Image analysis | Fully | Single | Deterministic | Episodic | Semi | Continuous |
| Part-picking robot | Partially | Single | Stochastic | Episodic | Dynamic | Continuous |
| Refinery controller | Partially | Single | Stochastic | Sequential | Dynamic | Continuous |
| Interactive English Tutor | Partially | Multi | Stochastic | Sequential | Dynamic | Discrete |
| Poker | Partially | Multi | Stochastic | Sequential | Static | Discrete |
| Backgammon | Fully | Multi | Stochastic | Sequential | Static | Discrete |



## Structure of Agent

Job of AI is to design an agent program that implements the agent function.

$$agent = archiecture + program$$

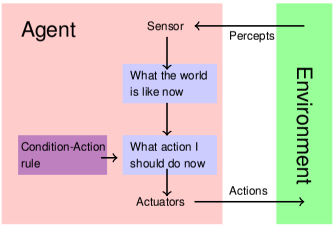
### Table driven agent

**Algorithm 1:** Table-Driven-Agent (percept)

```
append percept to percepts
action = LOOKUP(percepts,table)
return action
```

- Because of combinatorics, size of the table is an issue. Also it would be not be a nice idea to have table.
- Four basic kind of agents are: Simple reflex, model based, goal based, and utility based

## Simple Reflex Agent

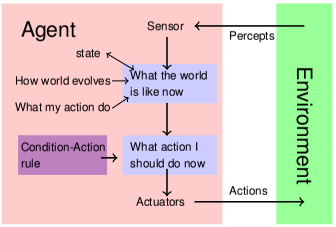



**Algorithm 2:** Simple-Reflex-Agent (percept)

1 state = INTERPRET-INPUT(*percept*)
2 rule = RULE-MATCH(*state*,*rules*)
3 action = rule.ACTION
4 **return** action



## Model Based Reflex Agent

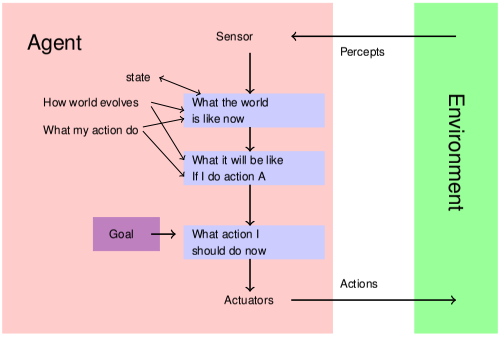



**Algorithm 3:** Model-Based-Reflex-Agent (percept)

1 state = UPDATE-STATE(*state*,*action*,*percept*,*model*)
2 rule = RULE-MATCH(*state*,*rules*)
3 action = rule.ACTION
4 **return** action



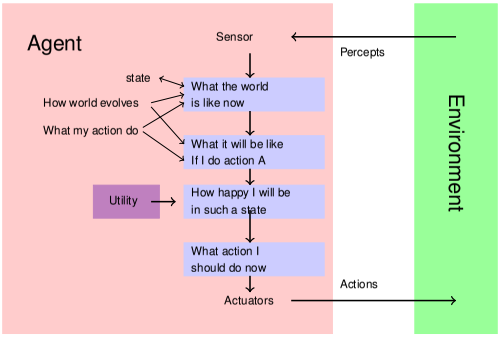
## Goal Based Agent




**Search** and **Planning** is needed. Consideration of future is important. Adaptive behavior change is possible.



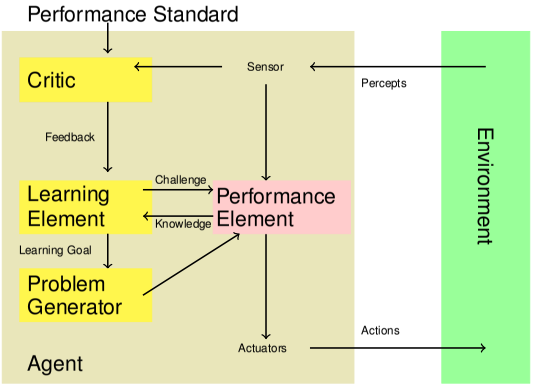
## Utility Based Agent




Utility evaluate how good it is. How cheap it is to reach the goal. Maximize **expected utility**. Trade-off between objectives could be managed. Can handle uncertainty.



## A General Learning Agent




**Critic** finds goodness of agent. **Learning element** make rules to improve or adapt. **Problem Generator** suggest experiments to test



## Problem-Solving Agent

Reflex agents cannot operate well if needs planning (or large table).

- A goal-based agent called **problem-solving agent** uses state of the world (cumulative)
- Uninformed search algorithms are not given any information about the problem other than its definition. They work, but may not efficiently (performance measure is always a concern for being intelligent).

**Consider an agent enjoying holiday in Arad, Romania**

What are performance measure? improve suntan, improve Romanian, sight seeing *etc.*

What if he has a flight from Bucharest next day?
*Adopt the goal of getting Bucharest on time.*

## Problem-Solving Agent

- **Goal** is some world-state.
- Agent's task is to find out how to act[2] now and in future.
- In our example[3] let three roads lead out of Arad, one towards Sibu, one to Timisoara, and one to Zerind. None of these achieves the goal. (he needs some familiarity with the geography of Romania *i.e.* environment)
- A map can specify the environment.
- Here environment is
  **observable** (agent always knows his current state)
  **discrete** (agent have finitely many actions to take)
  **known** (agent knows which action takes to which state)
  **deterministic** (each action has only one outcome)

Here solution to a problem corresponds to a fixed sequence of actions.

[2] Actions could be abstract, like goto A to B (not move 5 step, rotate 10 degree ...)
[3] Agent want to go to Bucharest



## Search

- The process of looking for a sequence of actions that reaches to goal is **search**
- How to look for, is an important question

**Problem is defined using five components**

1. The **initial state**: $in(Arad)$
2. Set of **actions**: $\{go(Subiu), go(Timisoara), go(Zerind)\}$
3. **Transition model**: $Result(in(Arad), go(Zerind)) = in(Zerind)$
4. **Goal test**: $\{in(Bucharest)\}$
5. **Path cost**: used to determine efficiency



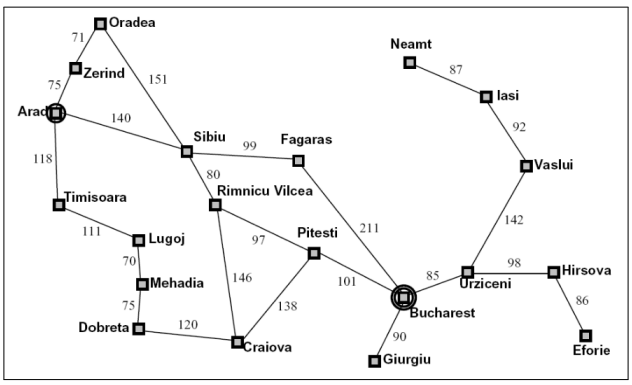
## Problem-Solving Agent

```
function SIMPLE-PROBLEM-SOLVING-AGENT(percept) returns an action
  inputs: percept, a percept
  static: seq, an action sequence, initially empty
          state, some description of the current world state
          goal, a goal, initially null
          problem, a problem formulation

  state ← UPDATE-STATE(state, percept)
  if seq is empty then do
     goal ← FORMULATE-GOAL(state)
     problem ← FORMULATE-PROBLEM(state, goal)
     seq ← SEARCH(problem)
  action ← FIRST(seq)
  seq ← REST(seq)
  return action
```



## Problem-Solving Agent






## Thank You!

**Thank you very much for your attention!**

**Queries ?**

(Reference[4])

[4] Book - *AIMA*, ch-02, Russell and Norvig.