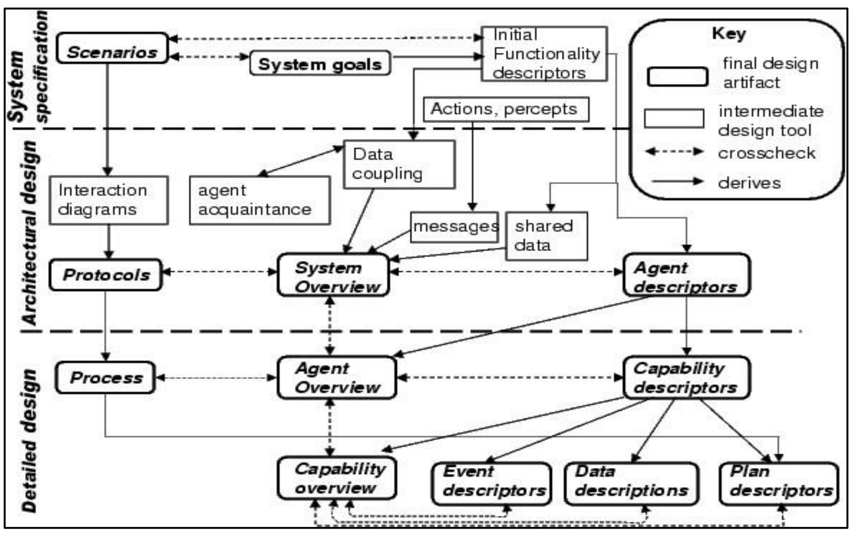
AI Midterm Revision

1. Automated Taxi Driver:

PEAS (Task Environment Description):

- Performance measure:
  - → safe, fast, legal, comfortable, maximize profits
- Environment:
  - → roads, other traffic, pedestrians, customers
- Actuators:
  - → steering, accelerator, breaks, signal, horn
- Sensors:
  - → cameras, sonar, speedometer, GPS

Intelligent Agent Design



- System Specification:
  - → Scenarios
  - → System goals
- Architectural design
  - → Protocols (peas)
  - → System overview (services)
  - → Agent descriptors
- Detailed Design
  - → Process (Actions)
  - → Agent overview (pseudocode)

Automated taxi driver scenario 1:

- **System Specification:**
  - → **Scenario**
    The hood open
  - → **System goals**
    Stop the car with avoiding accidents
- **Architectural design**
  - → **Protocols**
    Sensor, Breaks, Waiting light, Camera
  - → **System overview**
    - Sensor, Camera: to know that the hood is open or not.
    - Breaks: to stop the car
    - Waiting light: to mark that the car is going to park
  - → **Agent descriptors**
    If the hood is open, switch on the waiting light, decrease the speed, use break then park the car, use the driver responsible for closing the hood
- **Detailed Design**
  - → **Process**
    Move left/right, Stop, waiting light, Do nothing
  - → **Agent overview**

Function Hood_Problem_Solving_Agent (Percept) returns a state

If hood_open == true, then

Turn on the waiting light

Decrease the speed

Use the brake to decrease the speed and park the car

Use driver to close the hood

Else

Do nothing

Return closed

Automated taxi driver scenario 2:

- **System Specification:**
  - → **Scenarios**
    Flat tire
  - → **System goals**
    Stop the car with avoiding accidents
- **Architectural design**
  - → **Protocols**
    Sensor, Breaks, Waiting light, Hydraulic elevator.
  - → **System overview**
    - Sensor: to know that the air pressure in the tire
    - Breaks: to stop the car
    - Waiting light: to mark that the car is being gone to park
    - Hydraulic elevator: for lifting the car
  - → **Agent descriptors**
    If the tire is flat, switch on the waiting light, decrease the speed, use break then park the car, use the driver responsible for changing the tire
- **Detailed Design**
  - → **Process**
    Move left/right, Stop, waiting light, Do nothing
  - → **Agent overview**

```
Function Tire_Problem_Solving_Agent (Percept) returns a state
    If Flat_Tire == true, then
        Turn on the waiting light
        Decrease the speed
        Use the brake to decrease the speed and park the car
        Release the bag
        Get the spare tire
        Use the hydraulic elevator for lifting the car
        Use the Unscrew driver for the nail
        Change the tire
        Use the screw driver for the nail
        Use the hydraulic elevator for down the car
        Put the flat tire in the car
        Close the bag
    Else
        Do nothing
  Return Changed
```

Automated taxi driver scenario 3:

- **System Specification:**
  - → **Scenarios**

    Damage breaks
  - → **System goals**

    Stop the car with avoiding accidents
- **Architectural design**
  - → **Protocols**

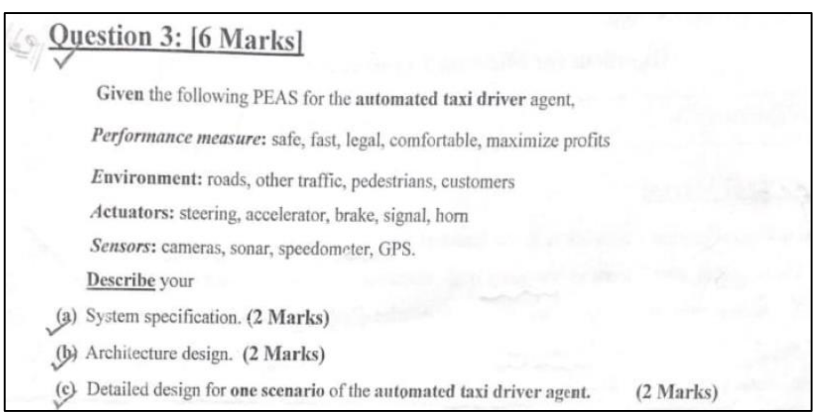
    Sensor, Waiting light.
  - → **System overview**
    - Sensor: to detect damage of break
    - Waiting light: to mark that the car will stop
    - Steering: to activate the handbrakes until taxi stops.
  - → **Agent descriptors**

    Sensor detects that the breaks are damaged, switch on the waiting light, and activate the handbrakes, and send an error message to the company.
- **Detailed Design**
  - → **Process**

    Move left/right, Stop, waiting light, Do nothing
  - → **Agent overview**

```
Function Break_Problem_Solving_Agent (Percept) returns a state
        If Break_Damage == true, then
             Turn on the waiting light
             Handbrake(on)
             SendErrorMessage("there is a problem in the breaks")
        Else
             Do nothing
   Return Checked
```

The following is a midterm question about automated taxi driver, and we answered them above.



Question 3: [6 Marks]

Given the following PEAS for the **automated taxi driver** agent,

Performance measure: safe, fast, legal, comfortable, maximize profits

Environment: roads, other traffic, pedestrians, customers

Actuators: steering, accelerator, brake, signal, horn

Sensors: cameras, sonar, speedometer, GPS.

Describe your

(a) System specification. **(2 Marks)**

(b) Architecture design. **(2 Marks)**

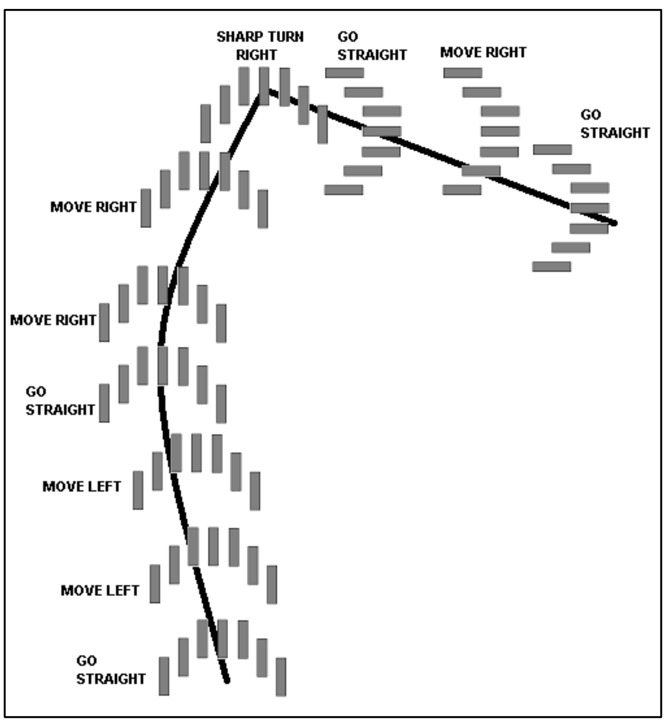
(c) Detailed design for **one scenario** of the **automated taxi driver agent.** **(2 Marks)**

2. Line Following Robot (LFR)

The LFR has the following requirements:

1. The robot must be capable of following a line.
2. It should be capable of taking various degrees of turns
3. It must be prepared of a situation that it runs into a territory which has no line to follow.
4. The robot must also be capable of following a line even if it has breaks.
5. The robot must be insensitive to environmental factors such as lighting and noise.
6. It must allow calibration of the line's darkness threshold.
7. Scalability must be a primary concern in the design.
8. The color of the line must not be a factor as long as it is darker than the surroundings.

LFR process involved



LFR Applications

Industrial automated equipment carriers, automated cars, tour guides in museums and other similar applications.

الشرح بالاسود لأي روبوت و الاحمر اجابه الاسأله

The following is a midterm question about LFR

Question 1: [10 Marks]

The robot could be defined as: "A reprogrammable, multifunctional manipulator designed to move materials, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks." In the essence of the practical case of **line Follower Robot** used for **Tour guides in museums:**

(a) What are these robots good at? **(1 Mark)**
(b) What other applications would you give to this robots? **(2 Mark)**
(c) What does building these robots teach us about humans? **(1 Mark)**
(d) How do these robots move? **(1 Mark)**
(e) What sensors might these robots have? **(2 Mark)**
(f) What are some problems with control of these robot actions? **(1 Mark)**
(g) How do you measures of performance of such robot? **(2 Mark)**

Solution

Robots are good at Repetitive tasks, Continuous operation, Complicated calculations, and Referring to huge databases.

(a) LFR used for tour guides in museum will be good at doing repetitive guides without getting tired or missing anything, not like humans.

Robots will be good at the following applications/tasks

- Dangerous → space exploration, chemical spill cleanup, disarming bombs, disaster cleanup
- Boring and/or repetitive → welding car frames, part pick and place, manufacturing parts.
- High precision or high speed → electronics testing, surgery, precision machining.

(b) Industrial automated equipment carriers, vacuum cleaner, and automated cars

Robots teach us about humans

- How do our sensors work? → eyes, brain
- How do we integrate sensors?
- How does our muscular-skeletal system work? → How do we grab and hold an object?
- How does our brain process information?
- What is nature of intelligence?
- How do we make decisions?

(c) Building LFR robots teach us about humans→ how our muscular-skeletal system work, how our sensors work, and how do we make decisions.

الاجابه تفصيليه جدا و مش لازم تبقى كده انا جاباها من النت..
حاولوا تفهموا الكونسبت في العموم و اكتبوا بطريقتكوا

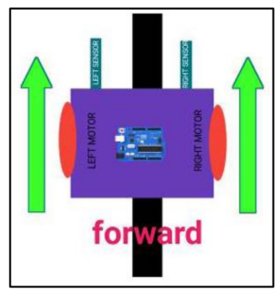
How does robots move

- Simple joints (2D)
  - → Prismatic — sliding along one axis
  - → Revolute — rotating about one axis
- Compound joints (3D)
  - → ball and socket = 3 revolute joints
  - → round cylinder in tube = 1 prismatic, 1 revolute
- Degrees of freedom = Number of independent motions
  - → 3 degrees of freedom: 2 translation, 1 rotation
  - → 6 degrees of freedom: 3 translation, 3 rotation

(d) A typical line follower robot has two sets of motors, let's call them left motor and right motor. Both motors rotate on the basis of the signal received from the left and the right sensors respectively. The robot needs to perform 4 sets of motion which includes moving forward, turning left, turning right and coming to a halt. The description about the cases are given below.

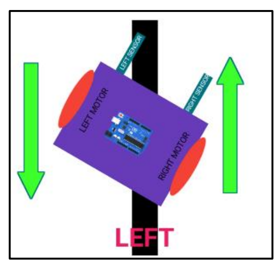
1. Moving forward:

In this case, when both the sensors are on a white surface and the line is between the two sensors, the robot should move forward.



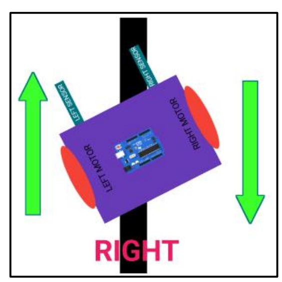
2. Turning Left:

In this case, the left sensor is on top of the dark line, whereas the right sensor is on the white part, hence the left sensor detects the black line and gives a signal, to the microcontroller. Since, signal comes from the left sensor, the robot should turn to the left direction. Therefore, the left motor rotates backwards and the right motor rotates in forward direction. Thus, the robot turns towards left side.



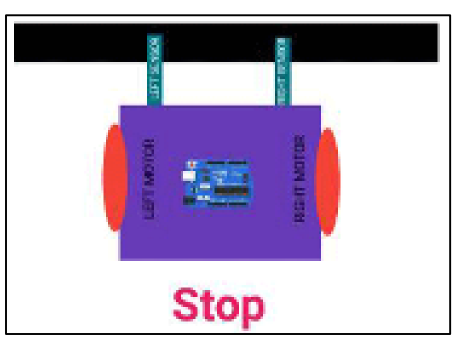
3. Turning Right:

This case is similar to the left case, but in this situation only the right sensor detects the line which means that the robot should turn in the right direction. To turn the robot towards the right direction, the left motor rotates forward and the right motor rotates backwards and as a result, the robot turns towards the right direction.



4. Stopping:

In this case, both the sensors are on top of the line and they can detect the black line simultaneously, the microcontroller is fed to consider this situation as a process for halt. Hence, both the motors are stopped, which causes the robot to stop moving.



Robots may use the following sensors

- Optical → Laser / radar, 3D, Color spectrum
- Pressure
- Temperature
- Chemical
- Motion & Accelerometer
- Acoustic → Ultrasonic

(e) LFR will need optical sensors to know where he is guiding, motion and accelerometer sensors to move wisely.

Some problems with control of robots actions

- Joint play, compounded through N joints.
- Accelerating masses produce vibration, elastic deformations in links.
- Torques, stresses transmitted depending on end actuator loads.
- Feedback loop creates instabilities → Delay between sensing and reaction.
- Firmware and software problems → Especially with more intelligent approaches

(f) LFR may have all the previous problems just as any robot.

To measure the performance of robot

- Speed and acceleration:
  - → Faster speed often reduces resolution or increases cost
  - → Varies depending on position, load.
  - → Speed can be limited by the task the robot performs (welding, cutting)
- Resolution (measure of movement):
  - → Often a speed tradeoff.
  - → The smallest movement the robot can make.
- Working volume (measure of theatre of operation):
  - → The space within which the robot operates.
  - → Larger volume costs more but can increase the capabilities of a robot
- Accuracy:
  - → The difference between the actual position of the robot and the programmed position.
  - → Repeatability
    Will the robot always return to the same point under the same control conditions?
    Varies depending on position, load?
- Cost: typically, a tradeoff – improved performance on other metrics vs lower cost

(g) LFR have the same measures.

اللي جاي ده ملخص عام للي اخدناه..

الدكتور كان قال اللي هيجي في الامتحان هو ال

Automated taxi driver و Line following robot.

و دول اتشرحوا في الصفحات اللي فاتت

Artificial intelligence as "the study and design of intelligent agents" where an intelligent agent is a system that perceives its environment and takes actions which maximize its chances of success

What is AI?

These are our four wings for an AI application:

1. Thinking humanly	The exciting new effort to make computers think machines with minds, in the full literal sense
2. Thinking rationally (Correct behaviour)	The study of the computations that make it possible to perceive, reason, and act
3. Acting humanly	The art of creating machines that perform functions that require intelligence when performed by people
4. Acting rationally	Computational intelligence is the study of the design of intelligence agents

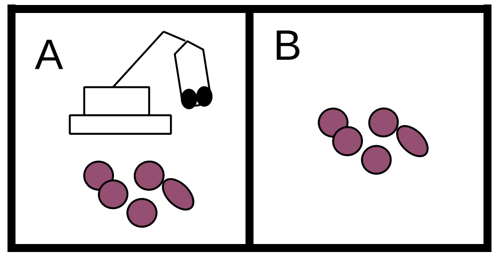
AI State of the art:

→ Deep Blue defeated the world chess champion Garry Kasparov in 1997
→ Proved a mathematical conjecture (Guising) unsolved for decades
→ No hands across America (driving autonomously 98% of the time from Pittsburgh to San Diego)
→ During the 1991 Gulf War, US forces deployed an AI logistics planning and scheduling program that involved up to 50,000 vehicles, cargo, and people
→ NASA's on-board autonomous planning program controlled the scheduling of operations for a spacecraft
→ Some programs solve crossword puzzles better than most humans

Agents: An agent is anything that can be viewed as perceiving its environment through sensors and acting through actuators

A Vacuum-cleaner agent

- → Percepts: location and contents, (Idealization: locations are discrete)
- → Actions: move, clean, do nothing (LEFT, RIGHT, SUCK, NOP)



- → Agent function:

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean]$, $[A, Clean]$	$Right$
$[A, Clean]$, $[A, Dirty]$	$Suck$
$\vdots$	$\vdots$

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*

Environment types:

1. **Fully observable (vs. partially observable):** An agent's sensors give it access to the complete state of the environment at each point in time.
2. **Deterministic (vs. stochastic):** The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic)
3. **Episodic (vs. sequential):** The agent's experience is divided into loosely connected modules (each episode consists of the agent perceiving and then performing a single action)
4. **Static (vs. dynamic):** The environment is unchanged while an agent is studying its decision about action. (The environment is semi dynamic if the environment itself does not change with the passage of time, but the agent's performance score does)
5. **Discrete (vs. continuous):** A limited number of distinct, clearly defined percepts and actions.
6. **Single agent (vs. multivalent):** An agent operating by itself in an environment.

Example: connect 4 game:

Environment Property	Description
Fully Observable vs. Partially Observable	The Connect Four environment is fully observable. The environment consists of the board, which has constant dimensions, and the pieces, which belong to either the player or the opponent. The agent has access to all of this information.
Deterministic vs. Stochastic	This environment could be considered deterministic, as there are no random elements at work here. The only unknown is the actions of the opponent. Therefore, the environment can be classified as strategic.
Episodic vs. Sequential	The environment could be either episodic or sequential, depending on the algorithm the agent uses. If the algorithm calls for random placement of a piece, then the environment is episodic. However, if the algorithm is more sophisticated, calling for prediction of the opponent's moves, then the environment is sequential.
Static vs. Dynamic	This environment is fully static. Time is not a factor in making the decision as to where to place pieces. Once it is the agent's turn, the state cannot be changed until it makes its move. The agent is also not penalized as a function of decision time.
Discrete vs. Continuous	Connect Four is a fairly simple game with a finite, albeit large, number of different states. Therefore, the environment is decidedly discrete.
Single agent vs. multi-agent	In this game, there are two agents at work. From the point of view of the AI agent, there is itself, and another agent. The other agent can either be a human player or another AI agent, which may or may not use the same algorithm. Since both agents (be they human or otherwise) are out to maximize their own performance measure and minimize their opponent's, the environment is classified as competitive multi-agent.

Agent types:

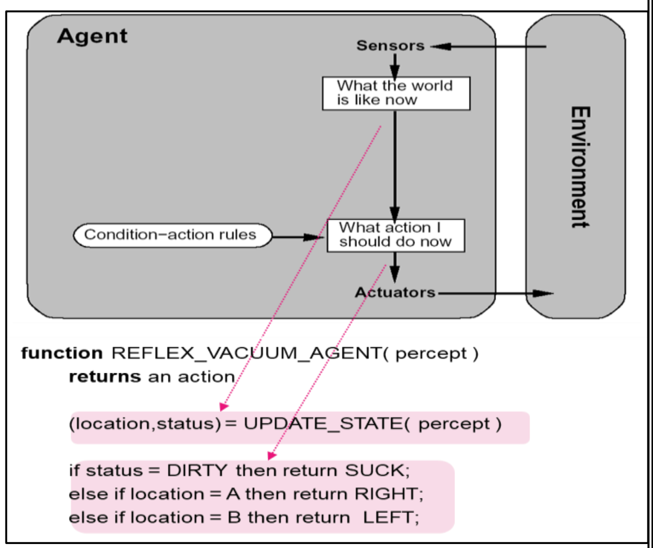
Russell & Norvig (2003) describe multiple types of agents and sub-agents as a first vision. For example:

- **Physical Agents** - A physical agent is an entity which percepts through sensors and acts through actuators (as a traditional agent).
- **Temporal Agents** - A temporal agent may use time based stored information to offer instructions or data acts to a computer program or human being and takes program inputs percepts to adjust its next behaviors.
- **Believable Agents** - An agent exhibiting a personality via the use of an artificial character for the interaction to practically express believes (believes against characters).

Some references classify agents into five classes based on their degree of perceived intelligence and capability:

- **Simple reflex agents**
  - → Acts only based on the current percept.
  - → The agent function is based on the condition-action rule:

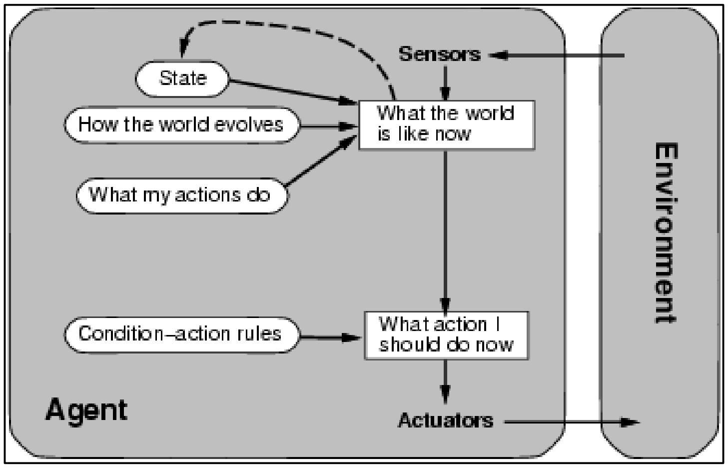
    if condition then action rule
  - → This agent function only succeeds when the environment is fully observable.
  - → Advantages
    - Selection of best action based only on current state of world and rules
    - Able to reason over past states of world
    - Still efficient, somewhat more robust



  - → Limitations and Disadvantages
    - No way to express goals and preferences relative to goals
    - Still limited range of applicability
  - → Implementation and properties
    - function ReflexAgentWithState (percept) returns action
    - static: state, description of current world state;

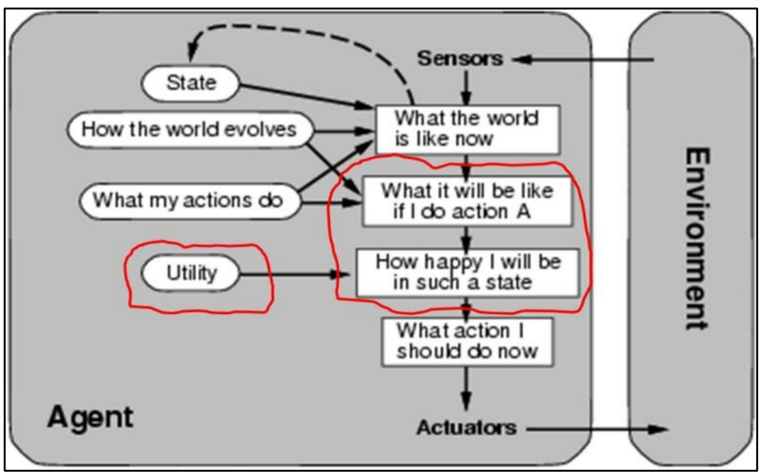
      rules, set of condition-action rules
    - state ← Update-State (state, percept)
    - rule ← Rule-Match (state, rules)
    - action ← Rule-Action {rule}
    - return action

➢ Model based reflex agents



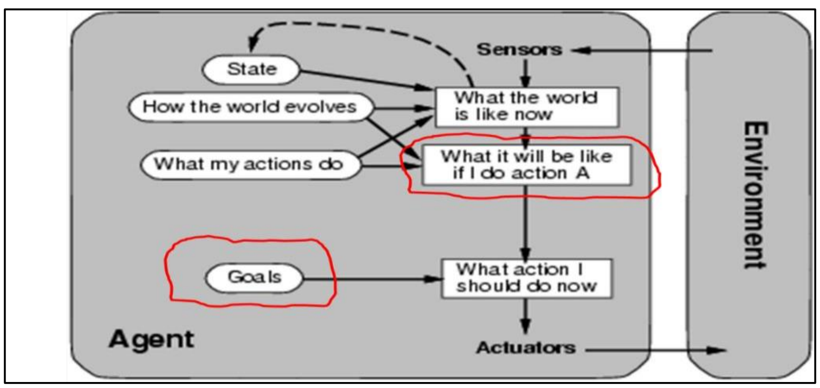
- Model-based agents can handle partially observable environments.
- Its current state is stored inside the agent maintaining structure which describes the part of the world which cannot be seen.
- This behavior requires information on how the world behaves and works. This additional information completes the “World View” model.

➢ Utility based agents

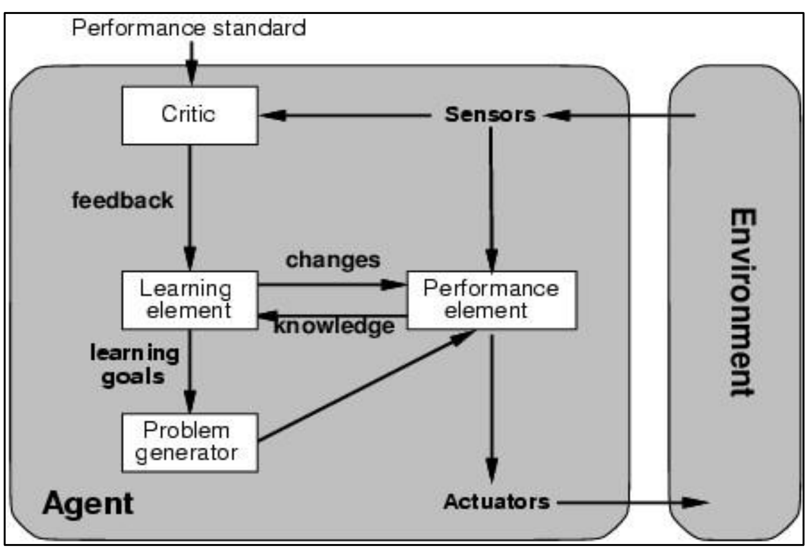


- Goal-based agents only distinguish between goal states and non-goal states.
- It is possible to define a measure of how desirable a state is.
- This measure can be obtained using a utility function which maps a state to a measure of the utility of the state.

- **Goal based agents**
    - → Goal-based agents are model-based agents which store information regarding situations that are desirable.
    - → This allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state.
    - → Implementation and Properties
      Functional description
        - Complete description using classical planning
        - Requires more formal specification
    - → Advantages
        - Able to reason over goal, intermediate, and initial states
        - Basis: automated reasoning
          One implementation: theorem proving (first-order logic)
          Powerful representation language and inference mechanism
    - → Limitations and Disadvantages
        - Efficiency limitations: can't feasible solve many general problems
        - No way to express preferences



- **Learning agents**
    - → The agent adapts its actions based on feedback (not only sensors)



Reasoning

- **Definition:** To reason is to draw conclusion appropriate to the situation.
- **Theories:**
  - → **<u>Deductive Reasoning:</u>**
    - Reasoning moves from a general principle to a specific conclusion.
    - Example:
      Premise: I will wash my car when the weather is good on weekends.
      Premise: Today is Sunday and the weather are hot
      Conclusion: Therefore, I will wash my car today.
  - → **<u>Inductive Reasoning:</u>**
    - Reasoning from the specific to the general.
    - Example:

      1= 12

      1+3 = 22

      1+3+5 = 32

      1+3+5+7 =42

      and, by induction $\sum$ (n successive odd integers) = n2
- **Applications:**
  - → Machine learning systems.
    Machine learning systems evolve their behavior over time based on experience.
    This may involve reasoning over observed events.
- **Problems and research areas:**
  - → Reasoning with Uncertainty
    - In many cases reasoning is done in an uncertain environment.
    - The reasoning methodologies need to be supplemented with procedures to handle the uncertainties.

Knowledge

- **Definition:** The field of artificial intelligence dedicated to representing information about the world in a form that a computer system can utilize to solve complex tasks.
- **Theories:**
  - → The Cognition:
    Transformation from Information to Knowledge
  - → The Decision-Making:
    Transformation from Knowledge to Intelligence
- **Applications:**
  - → Game playing
    You can buy machines that can play master level chess for a few hundred dollars. There is some AI in them, but they play well against people mainly through brute force computation.
- **Types:**
  - → Meta Knowledge: It’s a knowledge about knowledge and how to gain them.
  - → Procedural Knowledge: Gives knowledge about how to achieve something.
  - → Structural Knowledge: Describes what relationship exists between objects