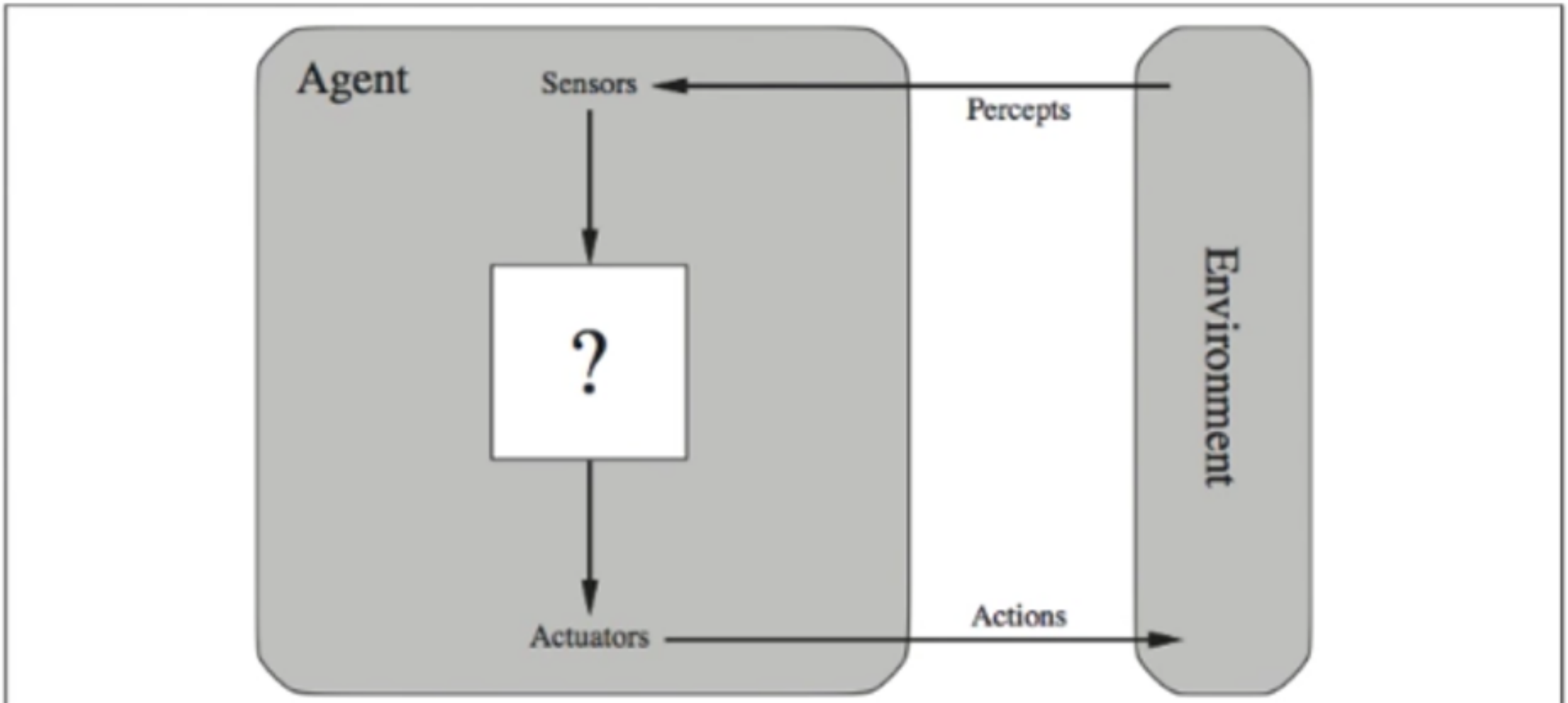
Chapter 2
Intelligent Agents

- **An agent is any thing that can perceive the environment through the sensors and acting upon the environment through actuators**

Intelligent Agents



Intelligent Agents
Considerations

- How should it act?
- How does it know it is doing what it is supposed to do?
- How does it know not to make a mistake?
- Should we monitor it at all times?
- What happens if it is put in a different environment?

Rational Agents
Doing the right thing

Performance Measure reports how “desirable” was an agent’s action.

Rationality depends on:

- The performance measure/criterion of success
- Agent’s prior knowledge of the environment
- The actions the agent can perform
- Agent’s percepts sequence to date

04-9: Rationality

- We want our agents to be *Rational* – that is, we want them to do the “right thing”
- What is the “right thing”? Performance measure
  - Condition or state of the world we want to achieve
  - Vacuum cleaner world: “Both rooms are clean” – could have additional criteria – minimize time or power consumption

04-10: Rationality

- Rationality is a specification of an *outcome*, rathen than a set of behaviors
- A rational agent tries to maximize its performance measure, given percepts and actions
- From text: 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

04-11: Rationality

- "expected" vs. actual – We don't require that our agent be able to predict the future, or predict unlikely events
- Information gathering might also be a rational action
  - Crossing the street without looking is irrational
- Rational agents must be able to *learn*, if they are situated in complicated enviornments
  - "Learning" == improving the agents performance int he current environment
  - Reducing uncertainty

04-12: Environments

- All agents exist in an environment
  - Software agents can exist in a software enviornment
- Task Environment
  - Performance measure
  - Environment
  - Actuators available to the agent
  - Sensors available to the agent

Task Environment
PEAS

- Agent Type
- Performance Measure
- Environment
- Actuators
- Sensors

- Agent Type: Taxi Driver
- Performance Measure: Safe, fast, comfortable trip, legal, $$
- Environment: Roads, traffic, customer, pedestrians, etc.
- Actuators: Steering, Gas pedal, brake, signal, horn,sign
- Sensors: Cameras, speedometer, GPS, odometer, etc.

Types of Environment of Agent

Intelligent Agents
Environment

- Fully Observable: Sensors provide all relevant aspects of env.
- Partially Observable: Some aspects of env. are not known

- Single agent
- Multi agent

- Static
- Dynamic

- Episodic
- Sequential

- Discrete
- Continuous

04-14: Observability

- Fully Observable: Agents sensors always give complete information
- Don't need to build a model of the world – can directly view it
  - Chess Playing: Fully observable
  - Bridge / Poker: Partially observable
  - Vacuum cleaner world: Parially observable (if it has only a local sensor – can't see the other room)

https://cs.gmu.edu/~sean/papers/LAMAS05Overview.pdf

04-18: Static / Dynamic

- Static:
  - World does not change while agent is thinking
  - No time pressure
  - Chess (mostly), classification
- Dynamic:
  - World changes while the agent is thinking
  - Agent must act within time pressure
  - Driving a car, most interactions in the physical world

04-17: Episodic / Seqential

- Episodic: Each action is independent
  - Perceive, decide, act. Repeat
  - Next decision does not depend on previous states
  - Don’t need to think ahead
  - Many diagnosis problems: Spam filter, image recognition, etc
- Sequential: Current decision effects future
  - Make a series of actions to accomplish a goal
  - Involves planning
  - Chess, driving a car

04-20: Discrete / Continuous

- Consider the state of the world, the percepts and actions of the agents, and how time is handled
- If the possible values are a discrete set, enviroment is descrete with repect to that characteristic
- If the values are continuously changing, environment is continuous

04-21: Discrete / Continuous

- Discrete
  - Chess, poker, backgammon
- Continuous
  - Image analysis (continuous sensor), car navigation, assembly robot (most all interactions with the physical world)

Deterministic v/s stochastic

If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise, it is stochastic.

Deterministic- Cross Word Puzzle

Stochastic- Poker

Intelligent Agents
Examples of Task Environments and Characteristics

Task Env.	Observable	Agents	Episodic	Static	Discrete
Chess	Fully	Multi	Sequential	static	discrete
Poker	Partially	Multi	Sequential	static	discrete
Image analysis	Fully	Single	Episodic	static	continuous
Butler Robot	Partially	Single	Sequential	dynamic	continuous

Types of Agents

Agents can be grouped into four classes based on their degree of perceived intelligence and capability :

Simple Reflex Agents
Model-Based Reflex Agents
Goal-Based Agents
Utility-Based Agents

Simple Reflex Agents

Suppose, an automated taxi driver agent is driving on road. The car in front of the taxi applying the brake and its brake lights are turned on and speed becomes slow.

In this situation agent takes the percepts from camera and other sensors and able to identify that current state of front car is applying the brake. So, it will take the decision to initiate breaking system to avoid collision.

Q. What is the world like now?

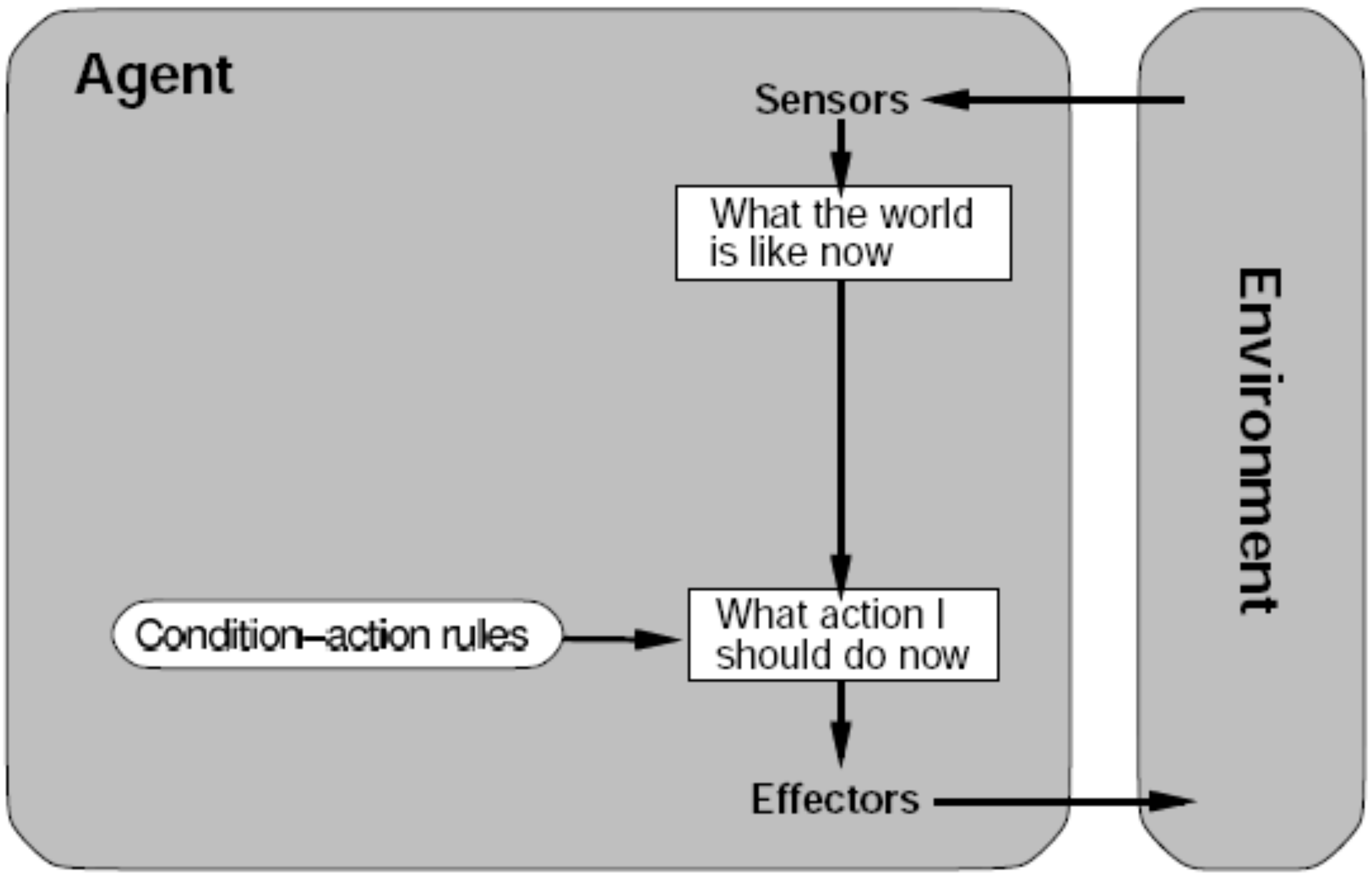
A. It's applying breaks.

Q. What action should I take?

A. Initiate breaking.

Condition Action Rules

Simple Reflex Agents




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

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

This agent selects actions based on the agents current perception or the world and not based on past perceptions.

For example if a mars lander found a rock in a specific place it needed to collect then it would collect it, if it was a simple reflex agent then if it found the same rock in a different place it would still pick it up as it doesn't take into account that it already picked it up.

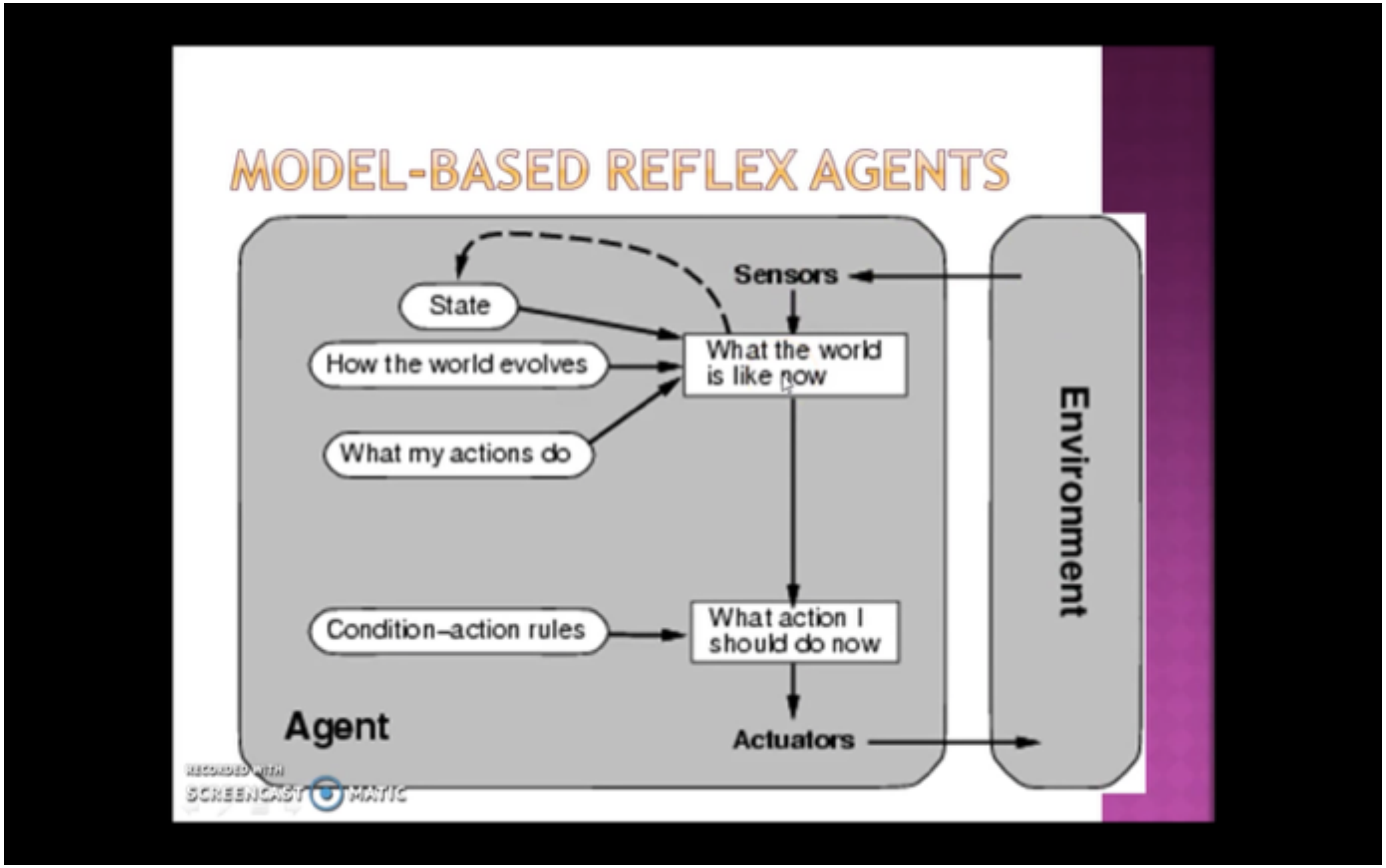
if hand is in fire then pull away hand

04-28: Reflex Agent

```
class ReflexVacuumAgent(Agent):
   def DoAction(location, status):
      if status == 'dirty':
         return 'suck'
      elif location == loc_A:
         return 'right'
      else:
         retrn 'left'
```

Intelligent Agents
Model Based Agents

- Partially Observable Problems
- Based on the input of the sensors and knowledge of the world
- State of the world updates based on inputs
- Rules are triggered.
- Smarts are shared. Still the coder can have a big influence.
- Maybe some non deterministic behavior here and there.



Model-based reflex agents are made to deal with partial accessibility; they do this by keeping track of the part of the world it can see now. It does this by keeping an internal state that depends on what it has seen before so it holds information on the unobserved aspects of the current state.

This time out mars Lander after picking up its first sample, it stores this in the internal state of the world around it so when it come across the second same sample it passes it by and saves space for other samples.

Another Example of Model Based Agent

- **The taxi may be driving back home, and it may have a rule telling it to fill up with gas on the way home unless it has at least half a tank.Although “driving back seem to an aspect of the world state, the fact of the taxi’s destination is actually an aspect of the agent’s internal state.**

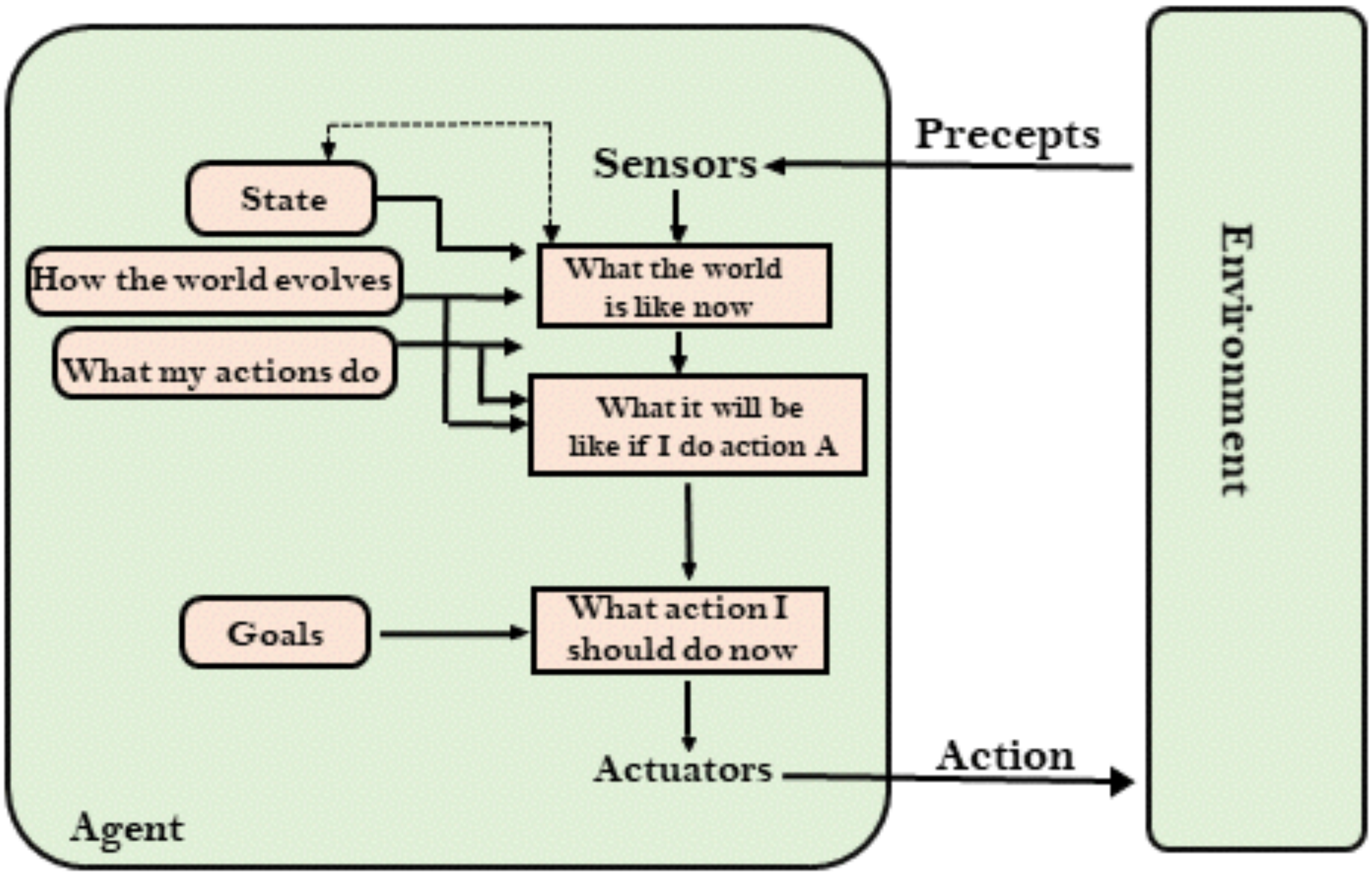
EXAMPLE-A MATCH REFEREE

- Percept
  - Points won by team
  - Misconduct
  - Timers
- Actions
  - Announce points won by team
  - Give warnings/yellow/red card
  - Review communications between team and third umpires
- Condition-action rule
  - If player(s) kick another player then misconduct
  - If misconduct level 1 then warning
  - If misconduct level 2 then yellow card
  - If misconduct level 3 then red card

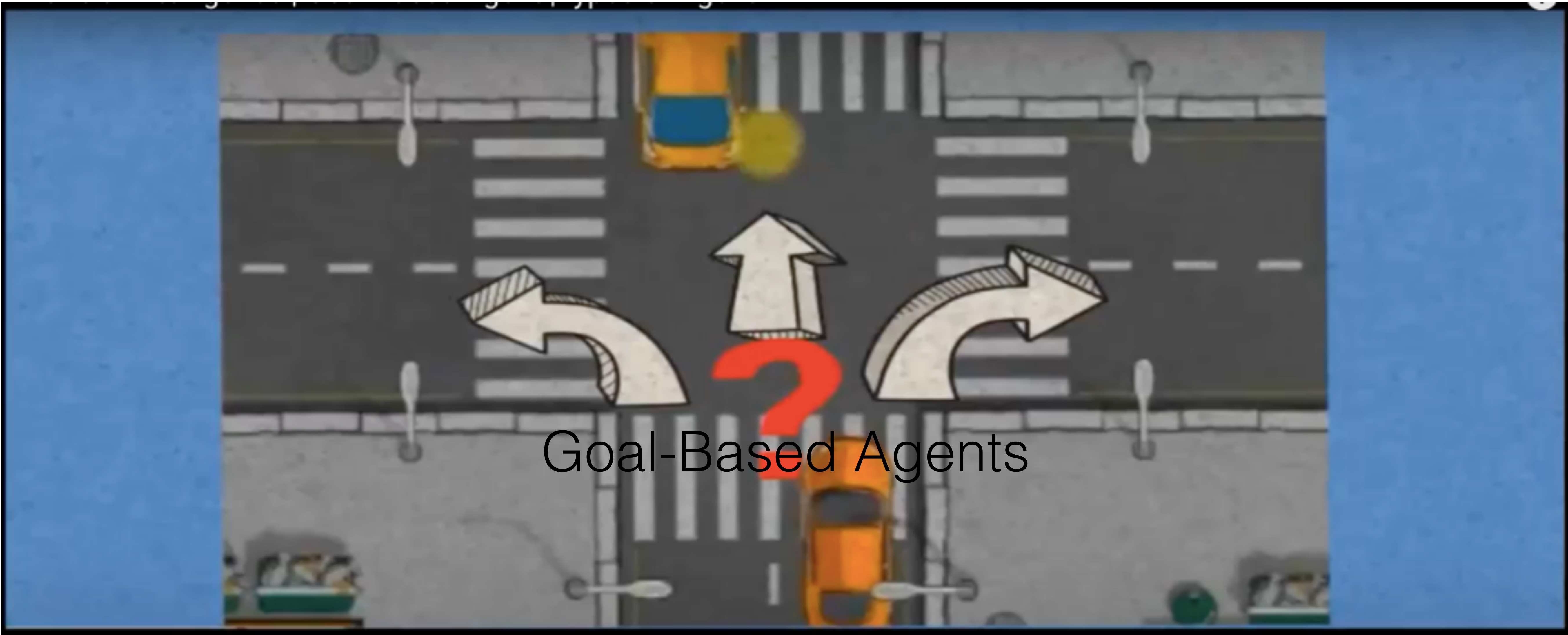
Intelligent Agents
Goal Based Agents

- Goals are evaluated by the system
- Based on the input of the sensors and knowledge of the world
- Goals are evaluated according to sensors and state
- Rules are triggered.
- Smarts are shared.
- Non deterministic behavior is more common

Goal-Based Agents



Goal-Based Agents Example



For example, at a road junction, the taxi can turn left, turn right, or go straight on.

The correct decision depends on where the taxi is trying to get to.

In other words, as well as a current state description, the agent needs some sort of **goal** information that describes situations that are desirable.

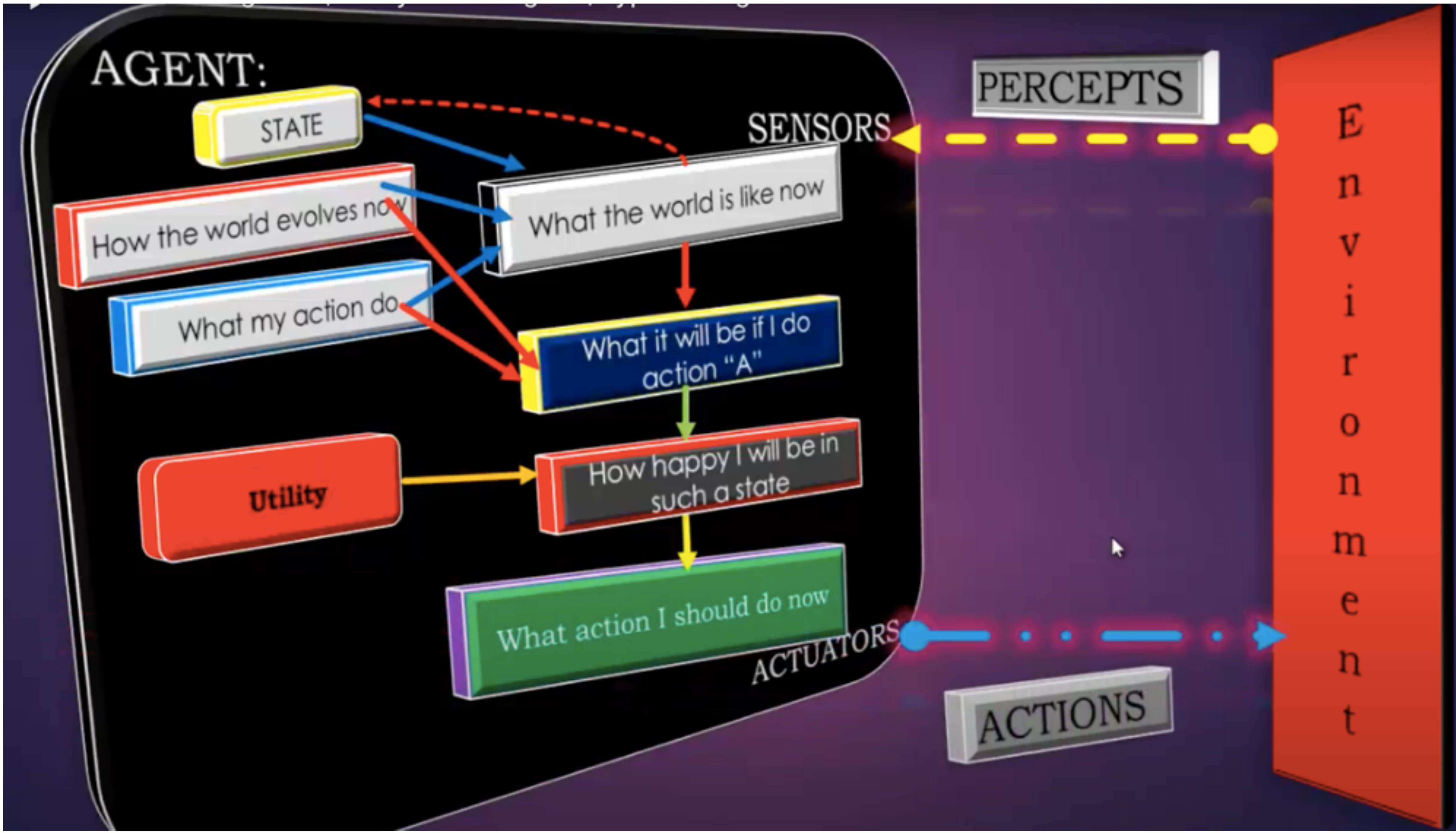
Utility Based Agent

Goals alone are not enough to generate high-quality behavior in most environments.

For example, many action sequences will get the taxi to its destination (thereby achieving the goal) but some are quicker, safer, more reliable, or cheaper than others.

Goals just provide a crude binary distinction between "happy" and "unhappy" states. If agent achieve its goal its in happy state if not its unhappy.

"Happy" does not sound scientific, economists, and computer scientist, So we are using the term "**Utility**" instead.


AGENT:
STATE
How the world evolves now
What my action do
What the world is like now
SENSORS
PERCEPTS
What it will be if I do action "A"
Utility
How happy I will be in such a state
What action I should do now
ACTUATORS
ACTIONS
Environment