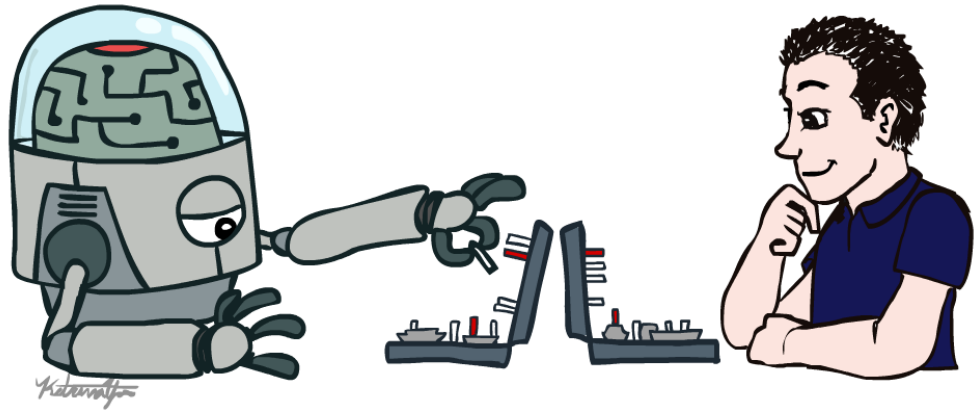
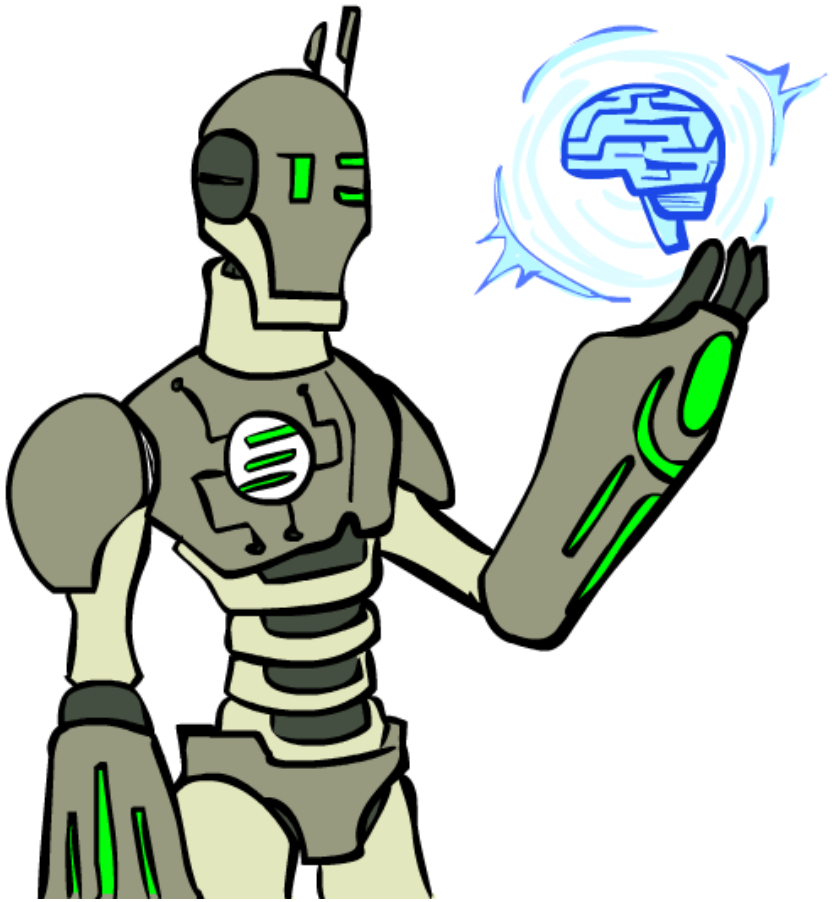
Lecture 01

Ashis Kumar Chanda
chanda@rowan.edu



RowanUniversity

Today



- Agents and environments
- Rationality
- Task Environment
- Environment types
- Agent types
- Examples

Agents

- An agent is anything that can be viewed as **perceiving** its environment through sensors and acting upon that environment through actuators.
- **Human agent:**

Sensors: eyes, ears, and other organs for sensors;

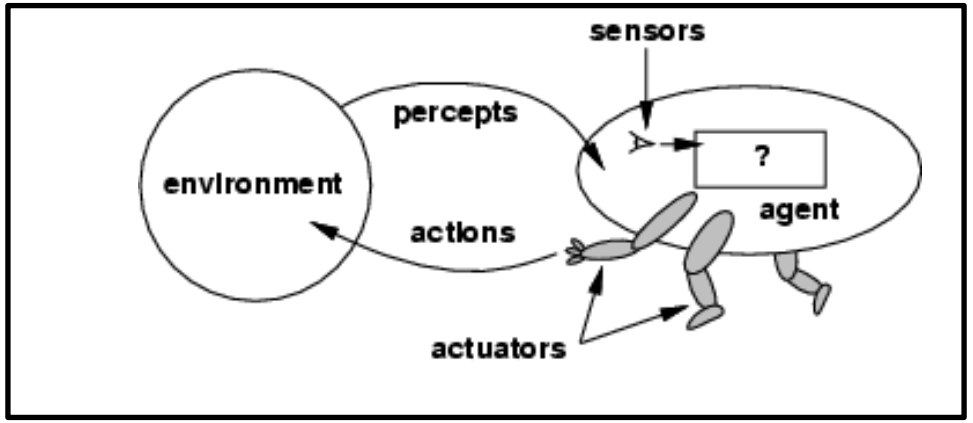
Actuators: hands, legs, mouth, and other body parts.

- **Robotic agent:**

Sensors: cameras, touch sensor and infrared range finders

Actuator: various motors for actuators

Agents and environments



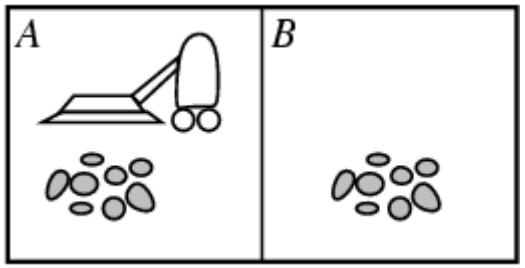
- The agent function maps from percept histories to actions:

$$[f: \mathcal{P}^* \rightarrow \mathcal{A}]$$

- The agent program runs on the physical architecture to produce *f*.
- An agent comprised of: architecture + program

Vacuum-cleaner world

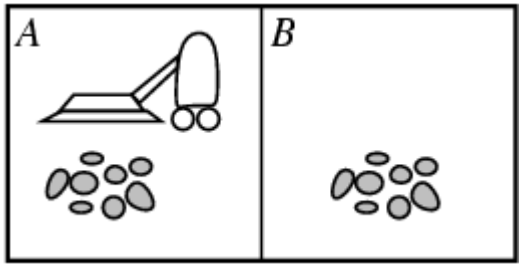
- A vacuum cleaner world with two locations, 1) clean, 2) dirty.
- The agent can move left or right and clean.



- Percepts: location and contents, e.g., [A, Dirty]
- Actions: *Left*, *Right*, clean, *NoOp*

A vacuum-cleaner agent

Percept Sequence	**Action**
[A, Clean]	Right
[A,Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	clean
….	…..



The next question is what makes an agent good or bad?

Rational agents (1/3)

- An agent should strive to "**do the right thing**", based on what it can perceive and the actions it can perform.
- The right action is the one that will cause the agent to be most successful.

- **Performance measure:** An objective criterion for success of an agent's behavior.

- E.g., performance measure of a vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.

Rational agents (2/3)

- **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.

Rational agents (3/3)

- What is rational an any time depends on four things:
  - The performance measure
  - The agent's prior knowledge of environment
  - The action that the agent can perform
  - The agent's percept sequence to date

- An agent is **autonomous** if its behavior is determined by its own experience (with ability to learn and adapt).

Task environment (1/5)

PEAS: Performance measure, Environment, Actuators, Sensors

Must first specify the setting for intelligent agent design.

Task environment (2/5)

- Consider, e.g., the task of designing an automated taxi driver:
  - **Performance measure**: Safe, fast, legal, comfortable trip, maximize profits.
  - **Environment**: Roads, other traffic, pedestrians, customers.
  - **Actuators**: Steering wheel, accelerator, brake, signal, horn.
  - **Sensors**: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard.

Task environment (3/5)

Agent: Medical diagnosis system

- **Performance measure:** Healthy patient, minimize costs.
- **Environment**: Patient, hospital, staff.
- **Actuators**: Screen display (questions, tests, diagnoses, treatments, referrals).
- **Sensors**: Keyboard (entry of symptoms, findings, patient's answers).

Task environment (4/5)

Agent: 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

Environment types

- **Fully observable** (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.
- **Deterministic** (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent.
- **Episodic** (vs. sequential): The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.

Environment types

- **Static** (vs. dynamic): The environment is unchanged while an agent is deliberating.
- **Discrete** (vs. continuous): A limited number of distinct, clearly defined percepts and actions.
- **Single agent** (vs. multiagent): An agent operating by itself in an environment. (i.e. solving a cross-world puzzle -> single agent, an agent playing chess is in a two agent environment)

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
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

- The **environment type** largely determines the **agent design**.
- The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent.

Agent functions and programs

- An agent is completely specified by the agent function mapping percept sequences to actions
- One agent function (or a small equivalence class) is rational.
- Aim: find a way to implement the rational agent function concisely.

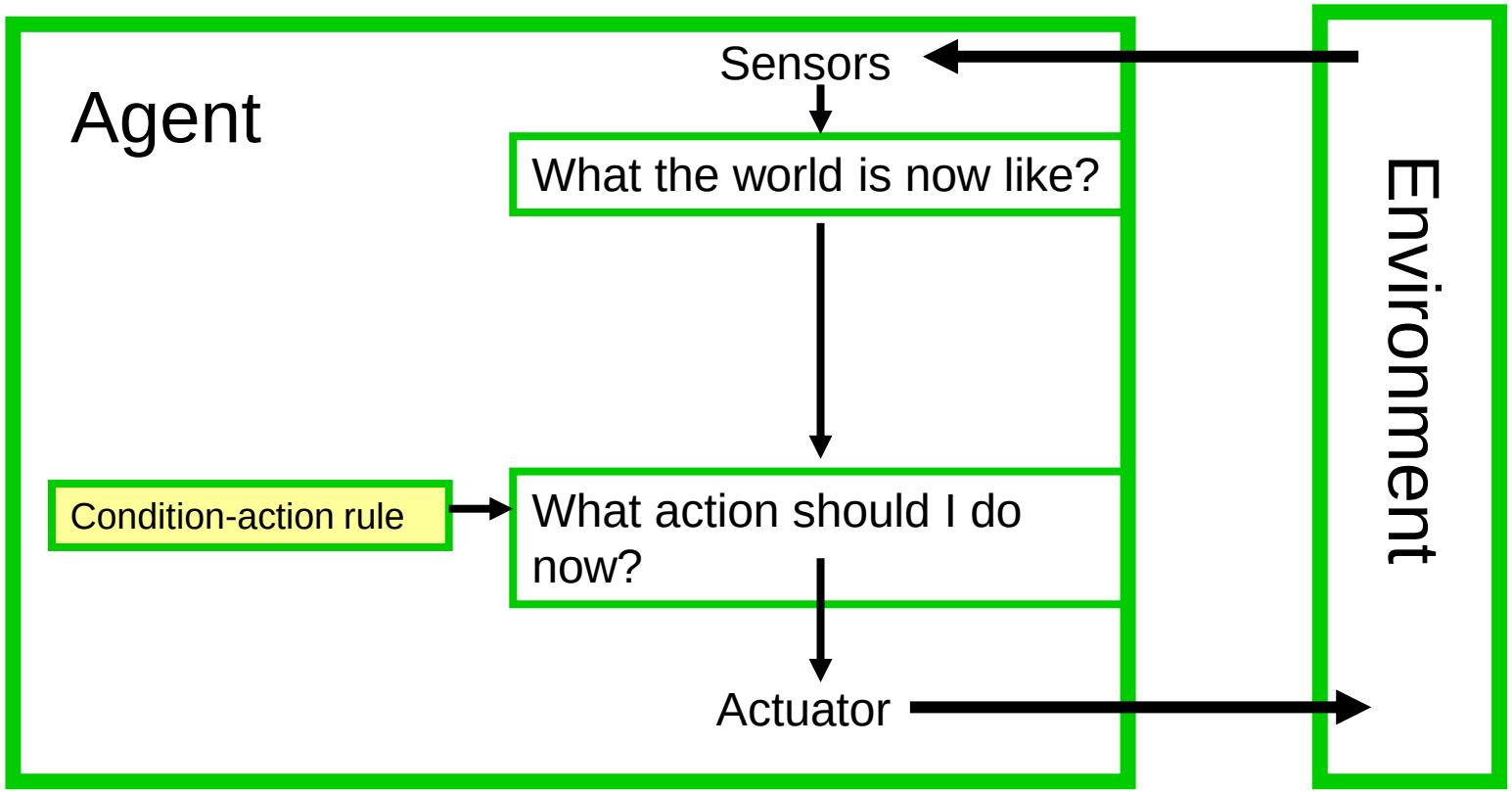
Agent types

Four basic types:

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

Simple reflex agents

- These agents select actions on the basis of the current percept, ignoring the rest of the percept history. Ex: vacuum cleaner



Agent program for a vacuum-cleaner agent

Function Reflex_Va_Agent([*location, status*])
return an action

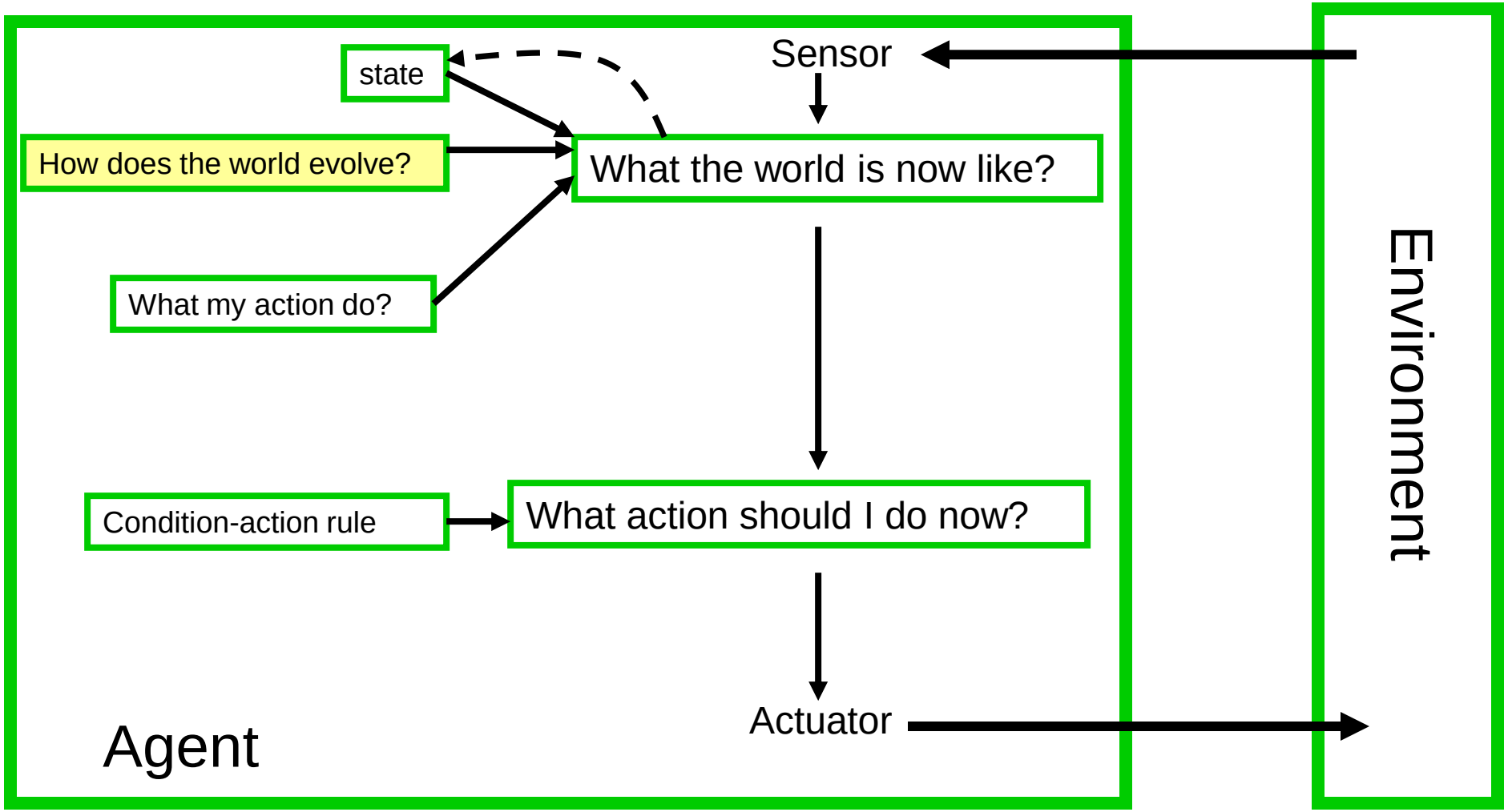
If *status* = *Dirty* then return *Suck*
Else if *location* = *A* then return *Right*
Else if *location* = *B* then return *Left*

Good only if environment is fully observable otherwise disaster.

For example, if location sensor is off then problem? [dirty] [clean]

Model-based reflex agents

Keep track of the part of the world it can't see now. Consider driving tasks such as lane changing.



Model-based reflex agents

Function Reflex_agent_with_state(percept)
return an action
Static: *state*, a description of the current world
rules, a set of condition action rules
action, the most recent action, initially none

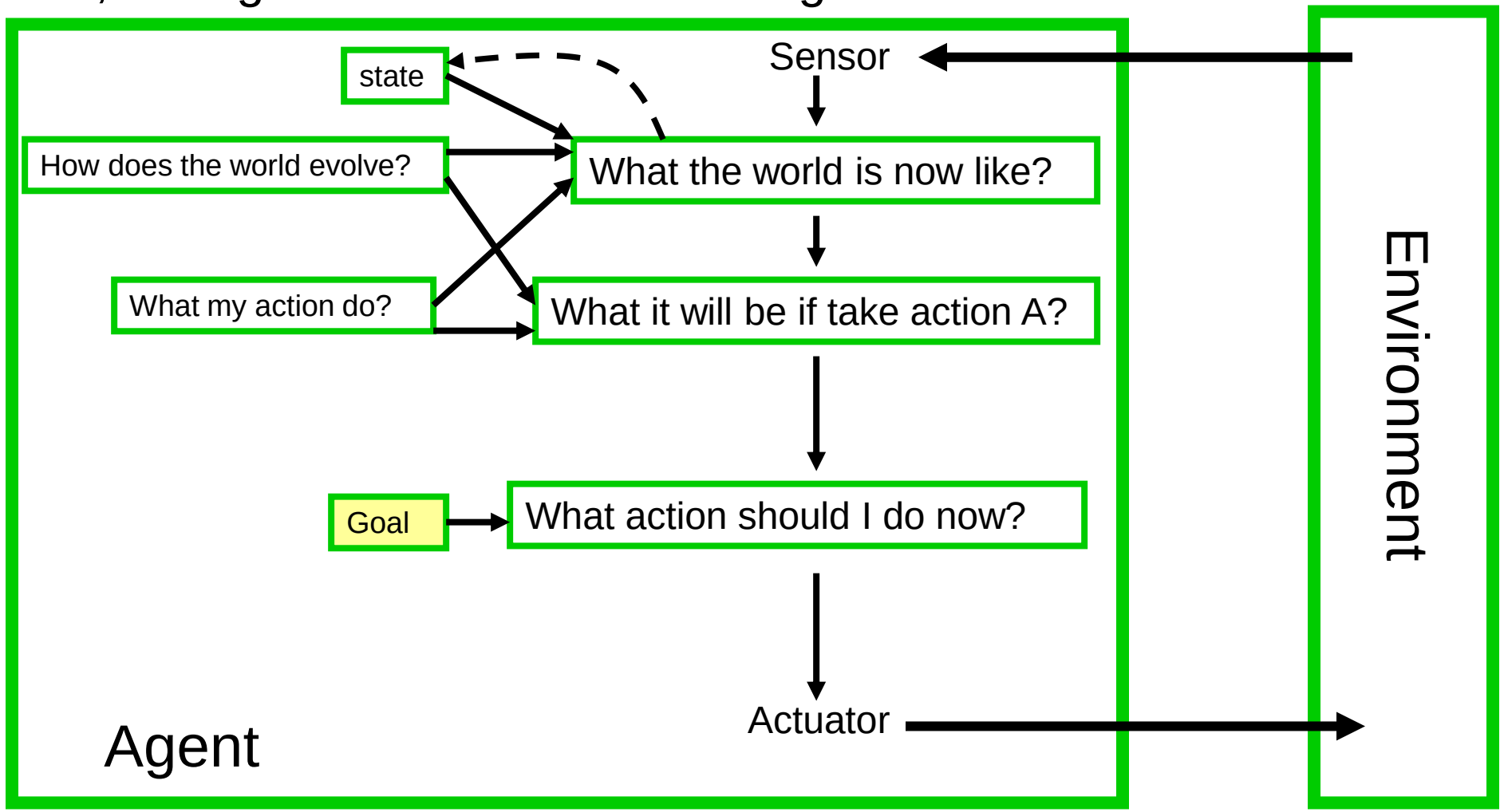
state ← Update-State (*state, action, percept*)
rule ← Rule-Match (state, rules)
action ← Rule-action (rule)
return *action*

Knowing about the current environment is not always enough to decide what to do

For example in the junction, left, right, go straight

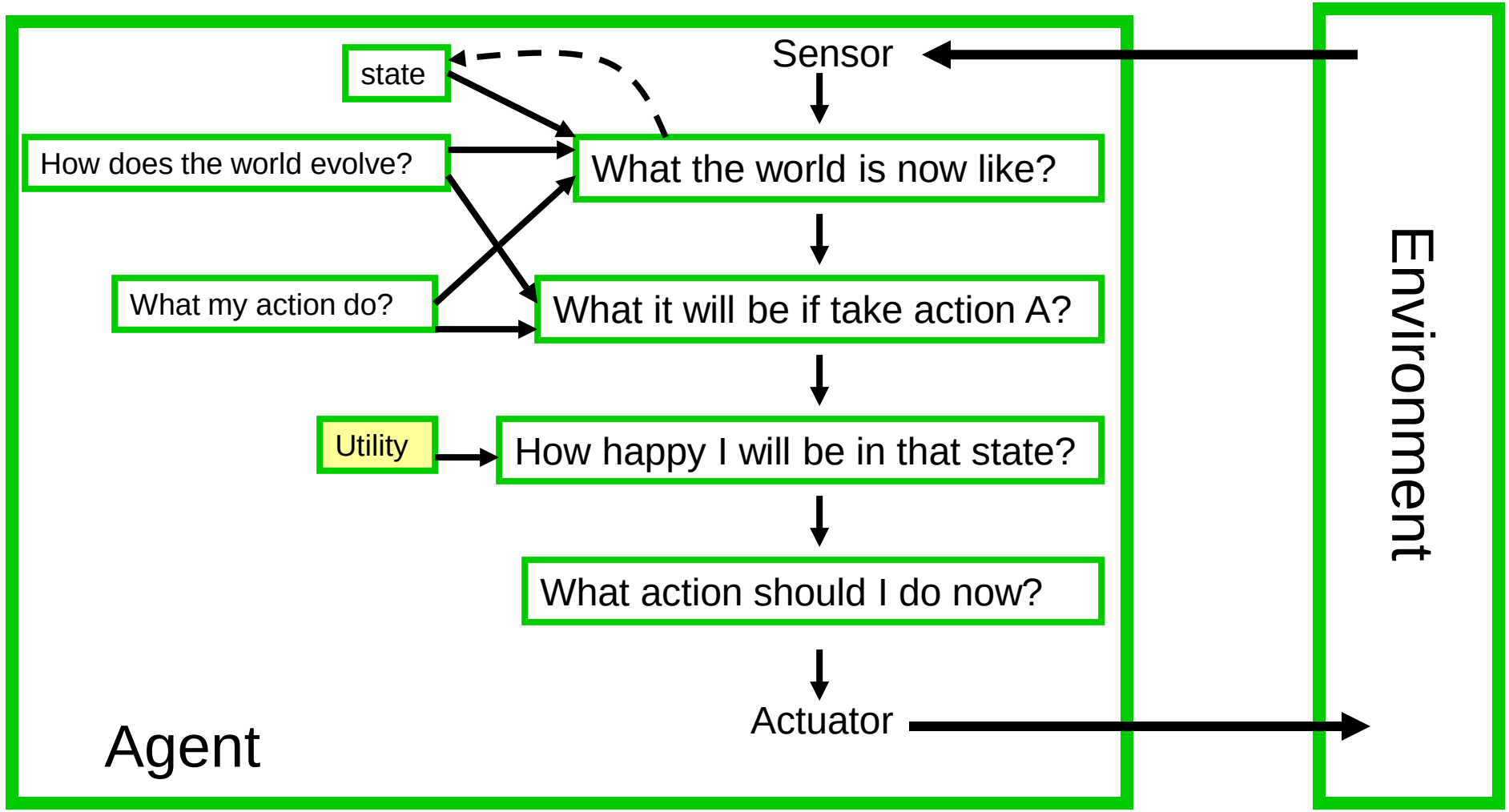
Goal-based agents

- In the junction problem, the correct decision depends on where the taxi is trying to get to. Or, the agent needs some sort of goal information.



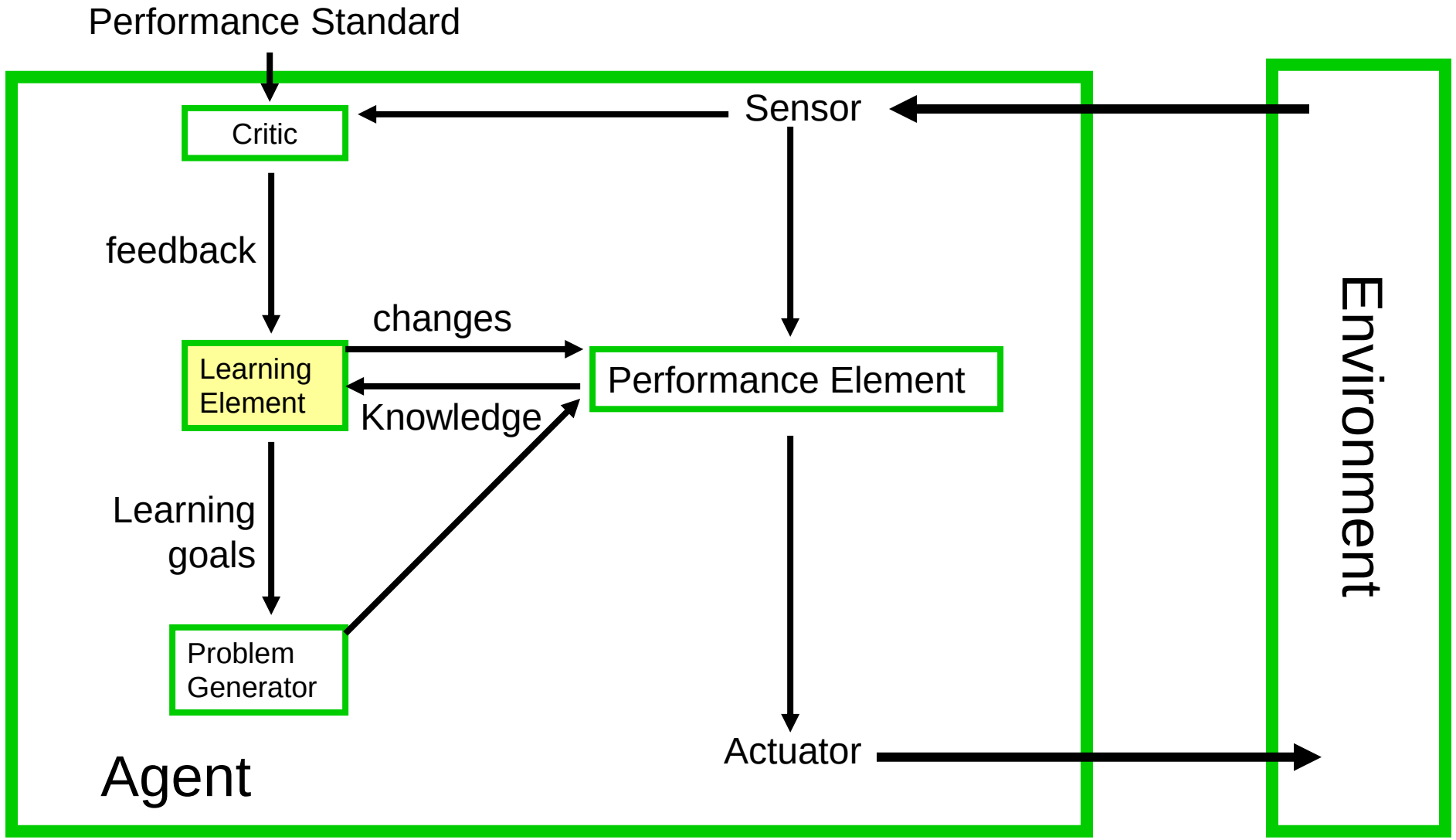
Utility-based agents

Goal alone is not enough. There can be many way to achieve goal but some are quicker, safer, more reliable or cheaper than other.



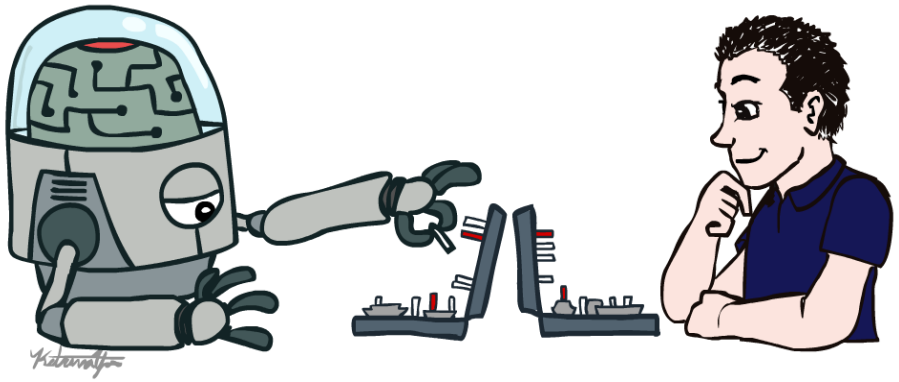
Learning agents

A learning agent can be divided into four conceptual components.



Next class?

- Solving problems by searching
- Uninformed or Blind search



Thanks!