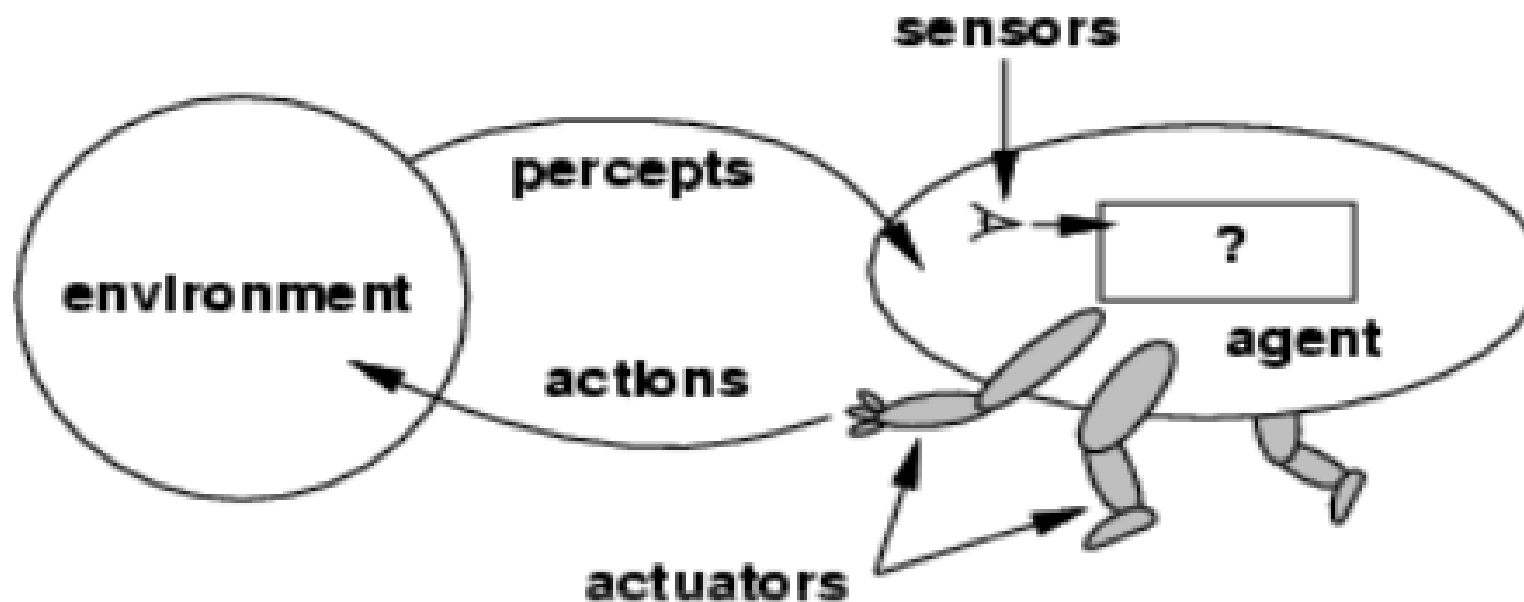


# **Chapter 4**

## **Intelligence Systems (Fundamental Concepts)**

---

# Tác tử



- Hàm tác tử: là hàm ánh xạ từ lịch sử nhận thức tới các hành động:  
 $f: P^* \rightarrow A$
- Chương trình tác tử: hoạt động (chạy) dựa trên kiến trúc thực tế của hàm  $f$
- Tác tử = Kiến trúc + Chương trình

# Tác tử

- Tác tử là bất cứ cái gì (con người, người máy, software robots, các bộ ổn nhiệt, ) ... có khả năng cảm nhận (nhận biết) môi trường xung quanh nó thông qua các bộ phận cảm biến (sensors) và hành động phù hợp theo môi trường đó thông qua các bộ phận hoạt động (actuators)
- Tác tử con người
  - Các bộ phận cảm biến: mắt, tai, và một số bộ phận cơ thể khác
  - Các bộ phận hoạt động: tay, chân, miệng, và một số bộ phận cơ thể khác
- Tác tử người máy
  - Các bộ phận cảm biến: các máy quay (cameras), các bộ truy tìm tín hiệu hồng ngoại
  - Các bộ phận hoạt động: các loại động cơ (motors)

# Tác tử và Môi trường



AGENT



ENVIRONMENT

**Environment:** the world in which the agent exists and operates.



AGENT



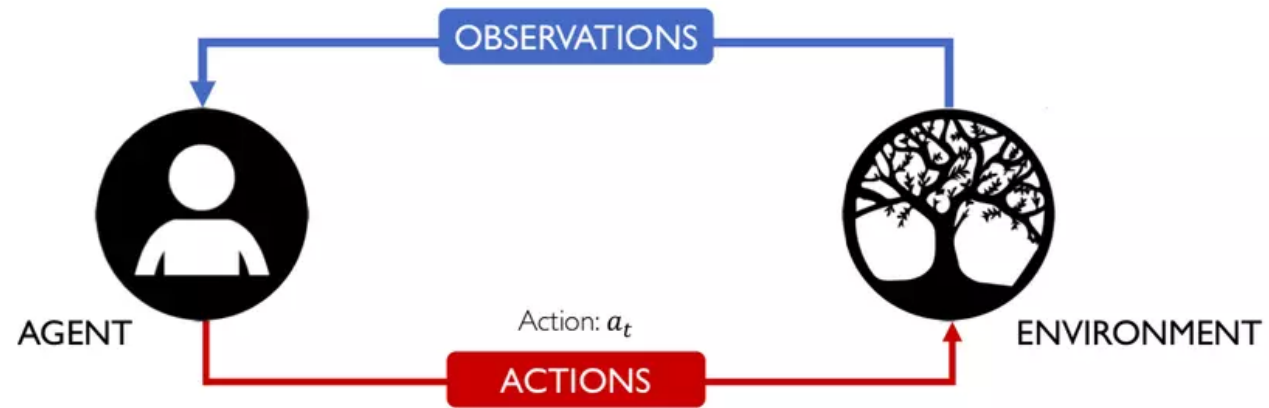
ENVIRONMENT

Action:  $a_t$

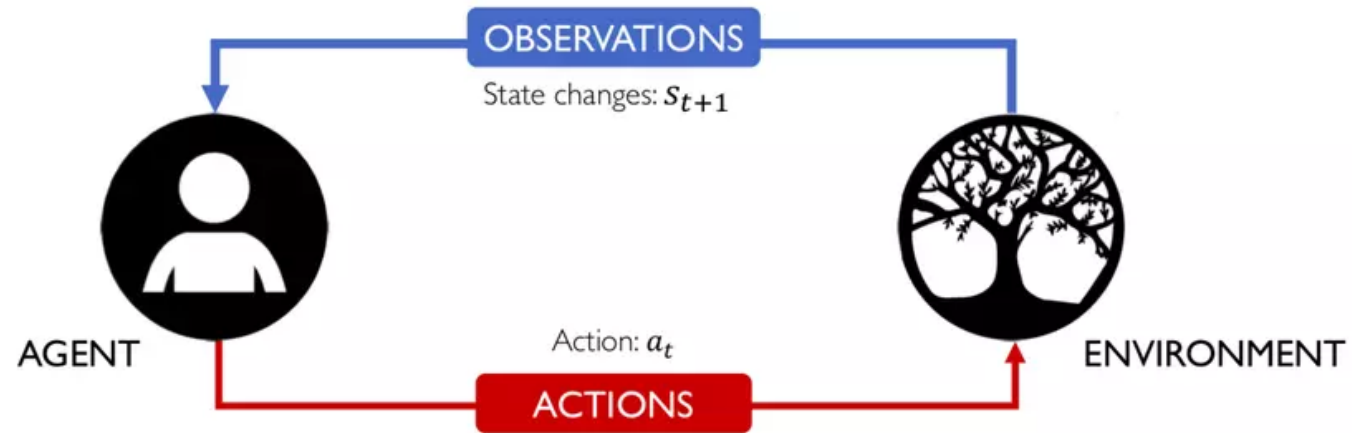
ACTIONS

**Action:** a move the agent can make in the environment.

# Tác tử và Môi trường



Observations: of the environment after taking actions.

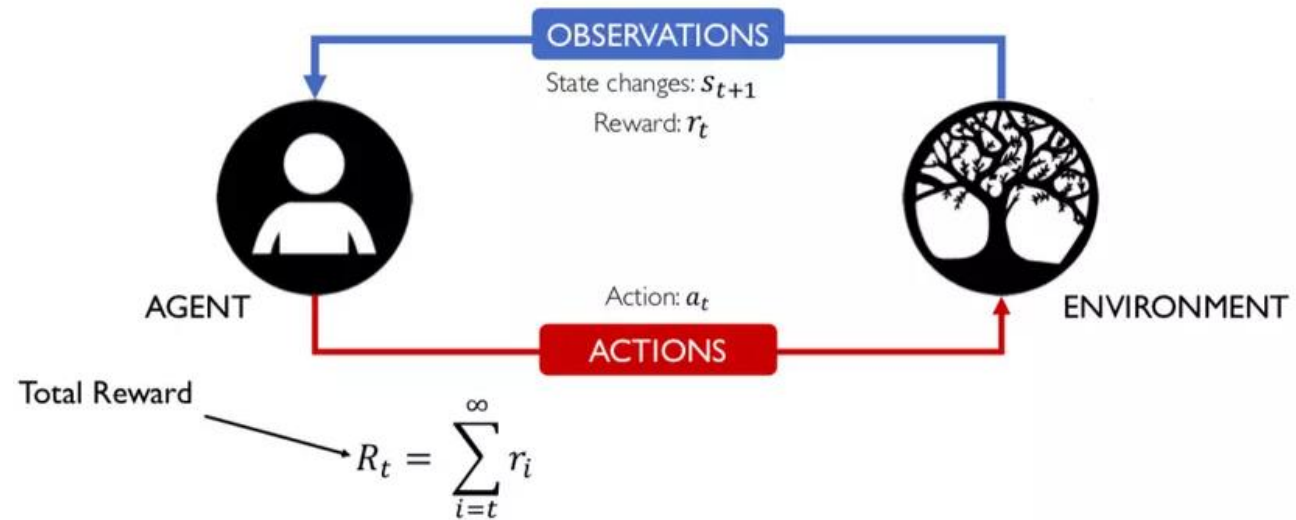


State: a situation which the agent perceives.

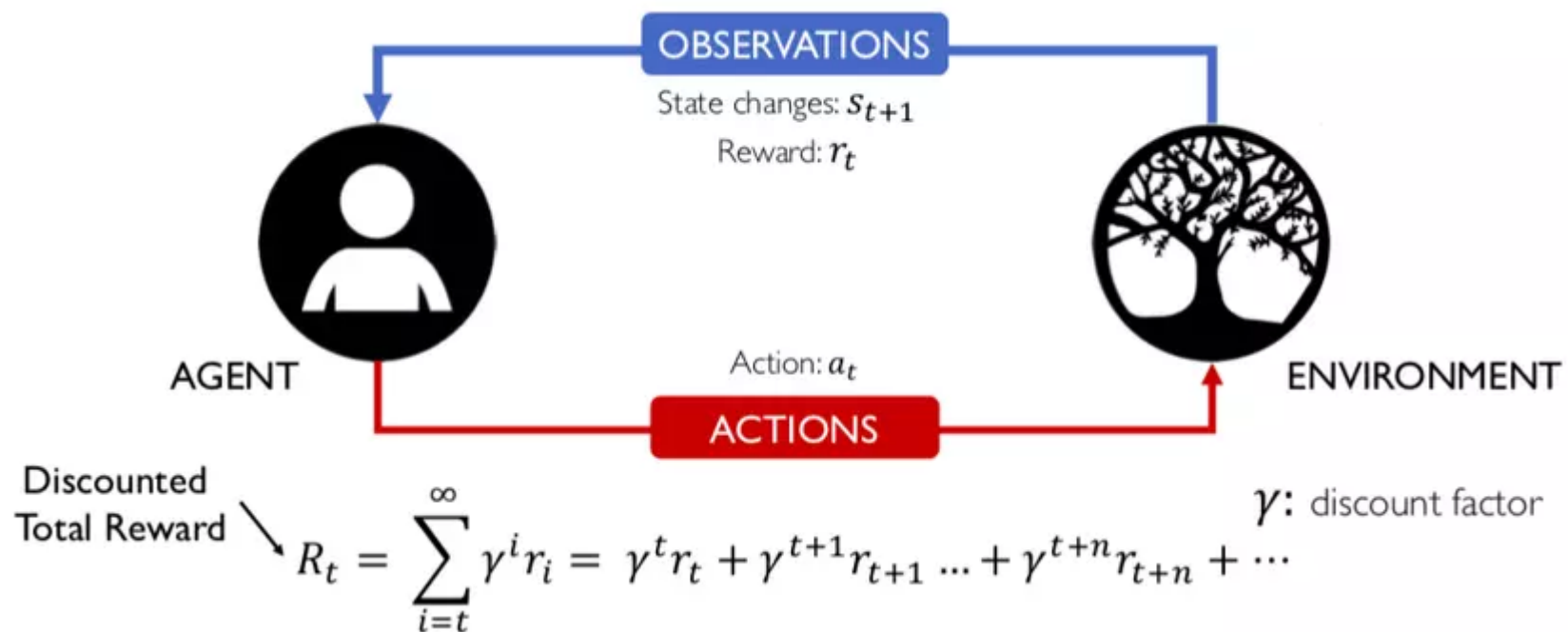
# Tác tử và Môi trường



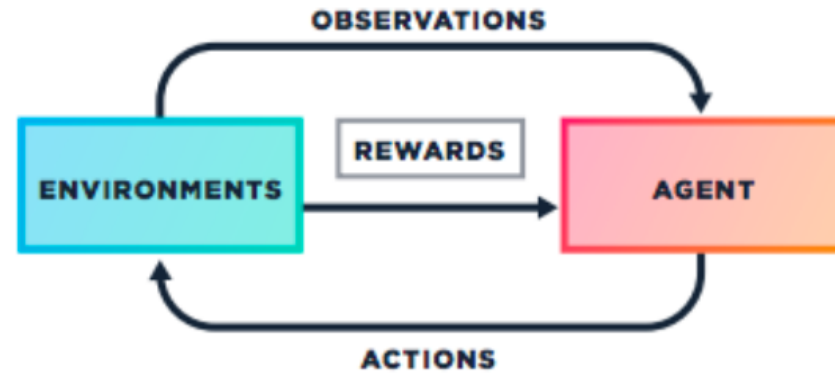
**Reward:** feedback that measures the success or failure of the agent's action.



# Tác tử và Môi trường



# Thiết kế hệ thống khen thưởng



Example: design reward system for chess playing agent.

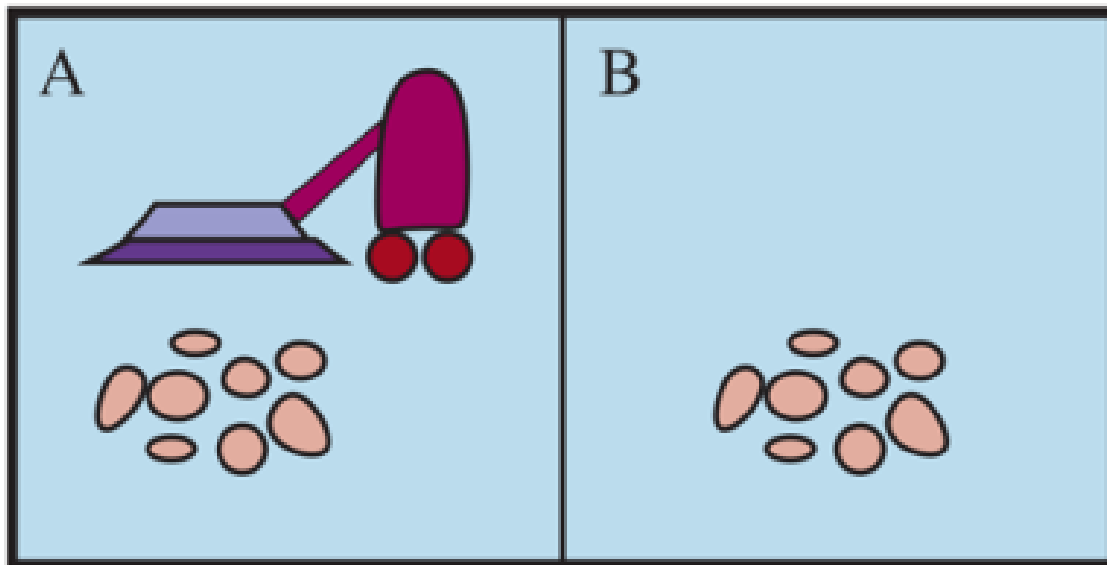
- System 1: *+1 for a win, 0 for a draw, -1 for a lose*
- System 2: *+1 for capturing an opponent's piece, -1 for being captured a piece*

Exercise: design reward system for maze escape agent.



# Tác tử và Môi trường

Example



**Percepts:** Vị trí (A or B), trạng thái (clean or dirty)

Ví dụ: [A, clean], [B, dirty]

**Actions:** *Left, Right, Suck, NoOp*

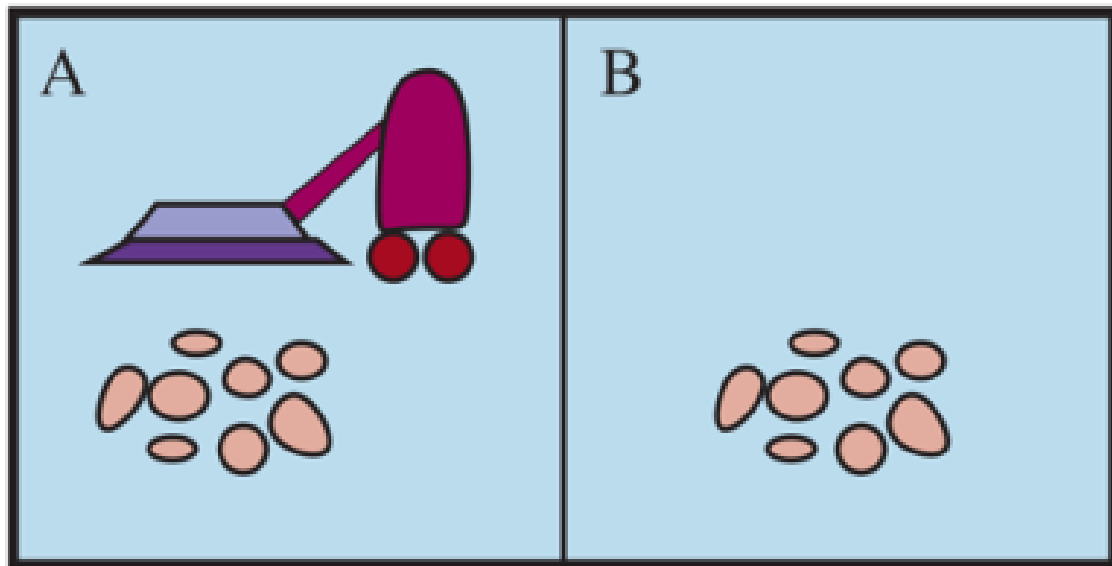
Thế giới của máy hút bụi chỉ có hai địa điểm. Mỗi địa điểm có thể sạch hoặc bẩn, và tác nhân có thể di chuyển sang trái hoặc phải và có thể dọn sạch ô vuông mà nó chiếm giữ.

Các phiên bản khác nhau của thế giới máy hút bụi cho phép các quy tắc khác nhau về những gì tác nhân có thể nhận thức, liệu hành động của nó có luôn thành công hay không, v.v.

Source: (Russell, 2016)

# Tác tử và Môi trường

- Example



**Percepts:** Vị trí (A or B), trạng thái (clean or dirty)

Ví dụ: [A, clean], [B, dirty]

**Actions:** *Left*, *Right*, *Suck*, *NoOp*

Percept sequence	Action
[A, Clean]	<i>Right</i>
[A, Dirty]	<i>Suck</i>
[B, Clean]	<i>Left</i>
[B, Dirty]	<i>Suck</i>
[A, Clean], [A, Clean]	<i>Right</i>
[A, Clean], [A, Dirty]	<i>Suck</i>
⋮	⋮
[A, Clean], [A, Clean], [A, Clean]	<i>Right</i>
[A, Clean], [A, Clean], [A, Dirty]	<i>Suck</i>
⋮	⋮

```
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
```

# Tại sao chúng ta cần Agent?

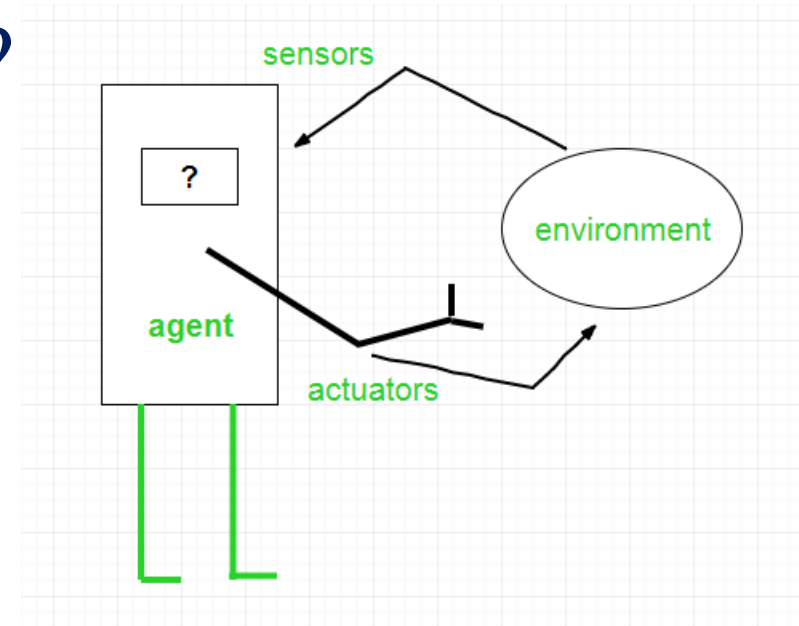
---

- Một công cụ để phân tích hệ thống.
- Mọi lĩnh vực kỹ thuật đều có thể được xem là thiết kế các thực thể tương tác với thế giới.
- AI thiết kế các thực thể có tài nguyên tính toán đáng kể và môi trường tác vụ đòi hỏi phải đưa ra quyết định không tầm thường.

# Tác tử hợp lý

## Rational agent

- Rational agent is one that does the right thing.
- Obviously, doing right thing is better than doing the wrong thing
- What does it mean to do the right thing?



# Tác tử hợp lý

---

Rational agent (tác tử hợp lý)

- Với mỗi chuỗi nhận thức có được,

⇒ tác tử hợp lý cần phải lựa chọn một hành động giúp cực đại hóa tiêu chí đánh giá hiệu quả hoạt động của tác tử đó,

⇒ dựa trên các thông tin được cung cấp bởi chuỗi nhận thức và các tri thức được sở hữu bởi tác tử đó

# Tác tử hợp lý

## Performance Measure

- Một quy tắc chung: Thiết kế các biện pháp hiệu suất theo
  - o Những gì người ta thực sự muốn trong môi trường
  - o Không phải cách người ta nghĩ tác nhân nên hành xử
- Ví dụ, trong thế giới máy hút bụi
  - o Chúng ta muốn sàn nhà sạch sẽ, bất kể tác nhân hoạt động như thế nào
  - o Chúng ta không hạn chế cách tác nhân hoạt động

# Tác tử hợp lý

---

**Performance Measure:** là tiêu chí cho sự thành công của actions của một agent.

Ví dụ: Tiêu chí đánh giá hiệu quả hoạt động của một tác tử máy hút bụi có thể là:

- mức độ làm sạch,
- thời gian hút bụi,
- mức độ điện năng tiêu tốn,
- mức độ tiếng ồn gây ra, ...

# Môi trường công việc

4 yếu tố cần cân nhắc khi thiết kế một tác tử tự động:

**P**erformance measure

**E**nvironment

**A**ctuators

**S**ensors

**PEAS:**

**Performance measure:** Tiêu chí đánh giá hiệu quả hoạt động

**Environment:** Môi trường xung quanh

**Actuators:** Các bộ phận hành động (For inputs)

**Sensors:** Các bộ phận cảm biến (For output)



# Môi trường công việc

4 yếu tố cân nhắc khi thiết kế một tác tử tự động:

**P**erformance measure

**E**nvironment

**A**ctuators

**S**ensors

Example: **PEAS - automated taxi driver**

**Performance measure:** Safe, fast, legal, comfortable trip, maximize profits, ...

**Environment:** Roads, other traffic, pedestrians, weather, ...

**Actuators:** Steering wheel, accelerator, brake, signal, horn, ...

**Sensors:** Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard, ...

Example: **PEAS - Medical diagnosis system**

**Performance measure:** Healthy patient, minimize costs, lawsuits, ...

**Environment:** Patient, hospital, staff

**Actuators:** Screen display (questions, tests, diagnoses, treatments, referrals)

**Sensors:** Keyboard (entry of symptoms, findings, patient's answers)

# Môi trường công việc

4 yếu tố cần cân nhắc khi thiết kế một tác tử tự động:

**P**erformance measure

**E**nvironment

**A**ctuators

**S**ensors

Example: **PEAS** - Thiết kế một tác tử nhật đồ vật

**Performance measure:**

**Environment:**

**Actuators:**

**Sensors:**

Example: **PEAS** - Thiết kế một tác tử dạy tiếng Anh tương tác

**Performance measure:**

**Environment:**

**Actuators:**

**Sensors:**

# Môi trường công việc

---

4 yếu tố cần cân nhắc khi thiết kế một tác tử tự động:

**P**erformance measure

**E**nvironment

**A**ctuators

**S**ensors

Example: **PEAS** - Thiết kế một tác tử lọc thư rác

**Performance measure:**

**Environment:**

**Actuators:**

**Sensors:**

# Environment types

- **Observable:** Cảm biến của tác tử cho phép nó tiếp cận toàn bộ trạng thái của môi trường
- **Single agent:** Một tác nhân hoạt động độc lập trong một môi trường.
- **Deterministic:** Trạng thái tiếp theo của môi trường được xác định hoàn toàn bởi trạng thái hiện tại và các hành động được tác tử lựa chọn
- **Episodic:** Trải nghiệm của tác tử được chia thành các “tập” độc lập, mỗi tập bao gồm phần tác tử nhận thức và sau đó là hành động.
- **Static:** Môi trường không thay đổi trong khi một tác nhân đang cân nhắc.
- **Discrete:** Một số lượng hạn chế các nhận thức và hành động riêng biệt, được xác định rõ ràng.
- **Known:** Trong một môi trường đã biết, kết quả (hoặc xác suất kết quả nếu môi trường là ngẫu nhiên) cho tất cả các hành động đều được đưa ra.

# Environment types

- Ví dụ:

	Chơi cờ tính giờ	Chơi cờ không tính giờ	Lái xe taxi
Quan sát đầy đủ?	có	có	không
Xác định?	chiến lược	chiến lược	không
Phân đoạn?	không	không	không
Tĩnh?	bán động	có	không
Rời rạc?	có	có	không
Tác tử đơn?	không	không	không

- Kiểu của môi trường có ảnh hưởng quyết định đối với việc thiết kế tác tử
- Môi trường trong thực tế thường có các đặc điểm: chỉ có thể quan sát được một phần, ngẫu nhiên, liên tiếp, thay đổi (động), liên tục, đa tác tử

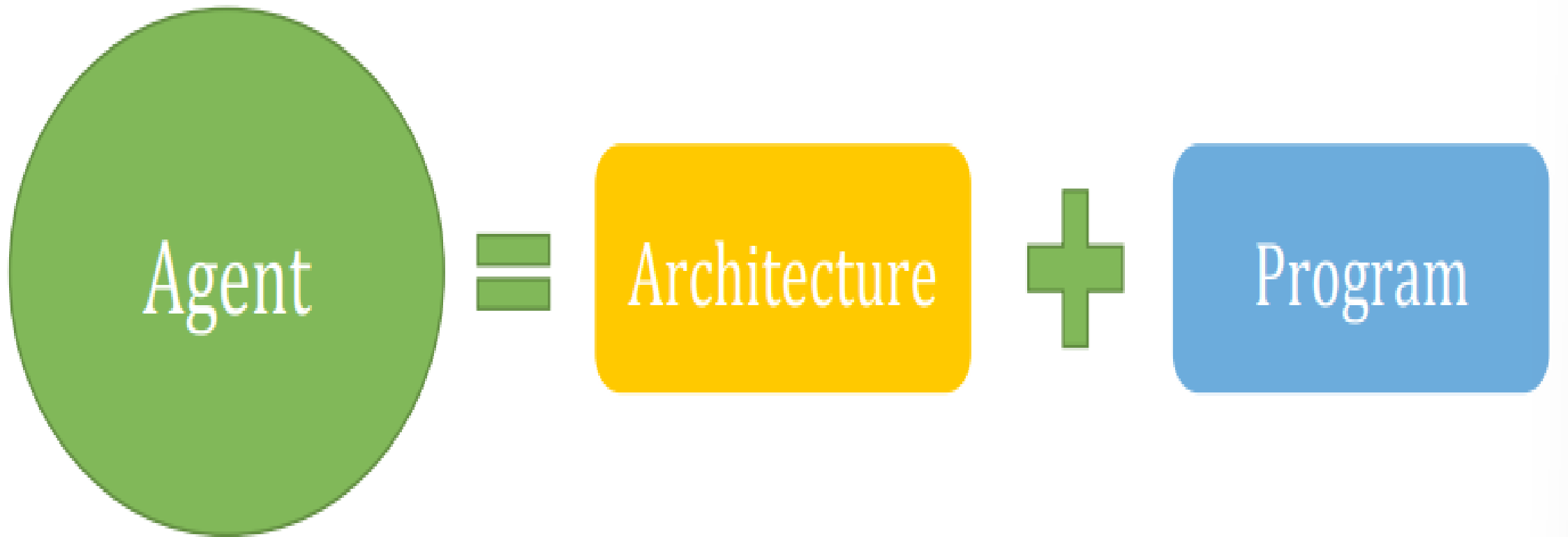
# Environment types

---

- Môi trường đơn giản nhất là:
  - Fully observable, deterministic, episodic, static, discrete and single-agent.
- Hầu hết các tình huống thực tế là:
  - Partially observable, stochastic, sequential, dynamic, continuous and multi-agent.

# Cấu trúc của các tác nhân

---



# Chương trình tác tử

---

- Đầu vào cho chương trình tác tử:
  - Chỉ cần percept hiện tại
- Đầu vào cho hàm tác tử
  - Toàn bộ percept sequence
  - Tác tử phải nhớ tất cả percept sequence
- Thực hiện một chương trình tác tử là:
  - Bảng tra cứu (look up table ) (hàm tác tử)



# Chương trình tác tử

- $P$  = the set of possible percepts
- $T$  = lifetime of the agent:
  - Tổng số percepts mà nó nhận được
- Kích thước của bảng Look-up
- Ví dụ: Xem xét trò chơi đánh cờ  $\sum_{t=1}^T |P|^t$ 
  - $P=10, T=150$
  - Bảng Look-up sẽ có ít nhất  $10^{150}$  entries

# Agent types

---

## **Five basic agent types:**

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

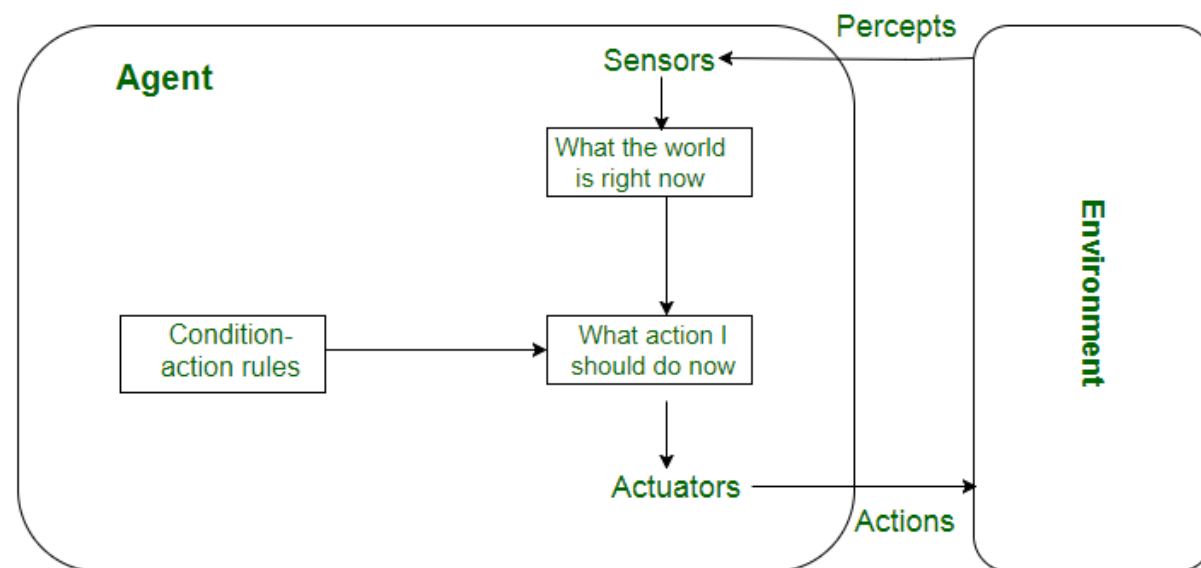
# Agent types

- Simple reflex agents

Tác tử phản xạ đơn giản:

→ Hành động theo một quy tắc (luật) có điều kiện phù hợp với trạng thái hiện thời (của môi trường)

```
function SIMPLE-REFLEX-AGENT(percept)  
static: rules (tập các luật có dạng: điều kiện-hành động)  
  
state ← INTERPRET-INPUT(percept)  
rule ← RULE-MATCH(state, rules)  
action ← RULE-ACTION[rule]  
return action
```



- Chỉ làm việc nếu môi trường là fully observable

# Agent types

- Model-based reflex agents

Tác tử phản xạ dựa trên mô hình:

- Sử dụng một mô hình nội bộ để giám sát trạng thái hiện tại của môi trường
- Lựa chọn hành động: giống như đối với tác tử phản xạ đơn giản

```
function REFLEX-AGENT-WITH-STATE(percept)
```

```
static: state (mô tả trạng thái hiện tại của môi trường)
```

```
       rules (tập các luật có dạng: điều kiện-hành động)
```

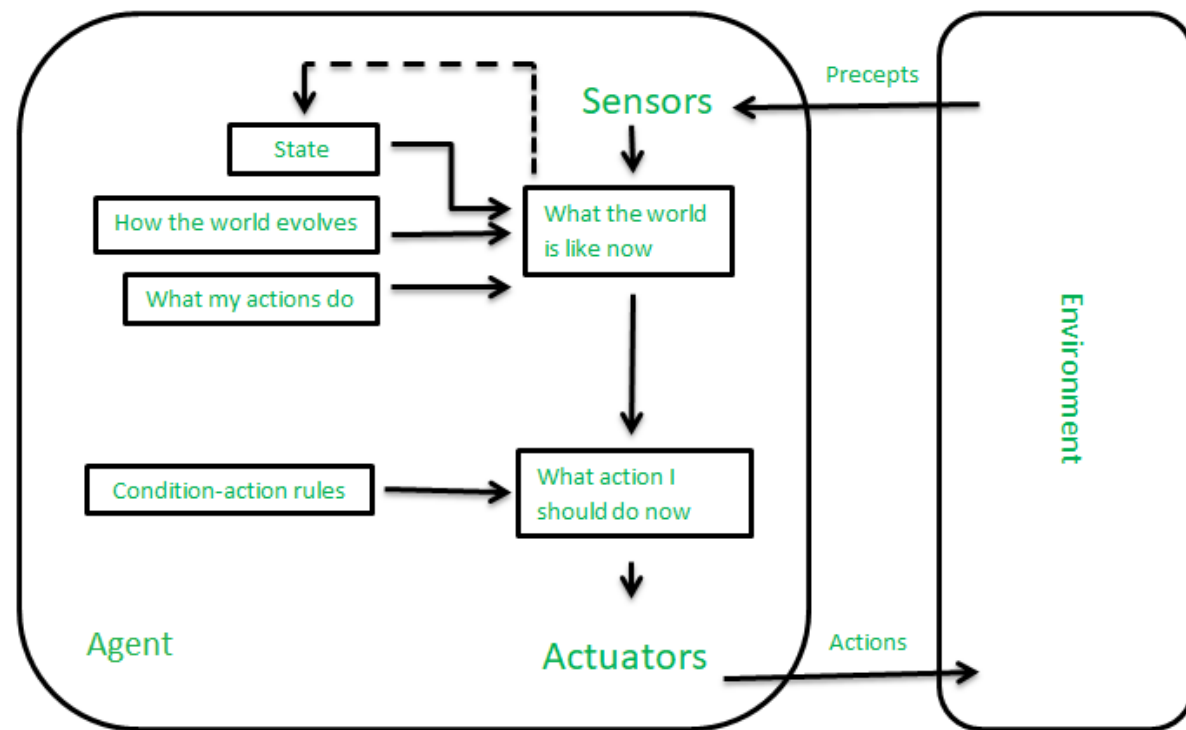
```
       action (hành động gần nhất)
```

```
state ← UPDATE-STATE(state, action, percept)
```

```
rule ← RULE-MATCH(state, rules)
```

```
action ← RULE-ACTION[rule]
```

```
return action
```



- Làm việc ở môi trường **partially observable**

# Agent types

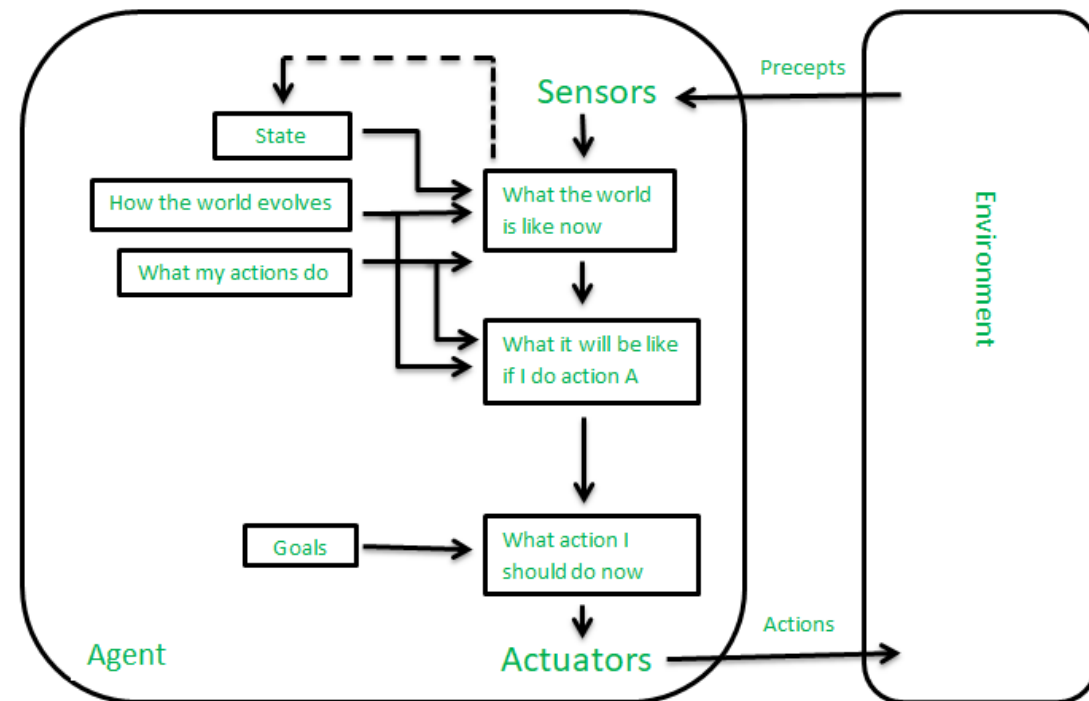
- Ví dụ bảng Agent với trạng thái nội bộ

If	Then
Nhìn thấy một vật thể phía trước, và rẽ phải, và bây giờ phía trước đã rõ ràng	Đi thẳng
Nhìn thấy một vật thể ở phía trước, rẽ phải và lại thấy vật thể ở phía trước	Dừng lại
Không nhìn thấy vật thể phía trước	Đi thẳng
Nhìn thấy một vật thể phía trước	Quay ngẫu nhiên

# Agent types

- Goal-based agents

- Biết về trạng thái hiện tại của môi trường: chưa đủ → Cần biết thêm thông tin về mục tiêu
  - Trạng thái hiện tại của môi trường: Ở một ngã tư, xe taxi có thể rẽ trái, rẽ phải, hoặc đi thẳng
  - Thông tin về mục tiêu: xe taxi cần đi tới đích đến của hành khách
- Tác tử dựa trên mục tiêu
  - Theo dõi trạng thái hiện tại của môi trường
  - Lưu giữ một tập các mục tiêu (cần đạt được)
  - Chọn hành động cho phép (rốt cuộc) sẽ đạt đến các mục tiêu

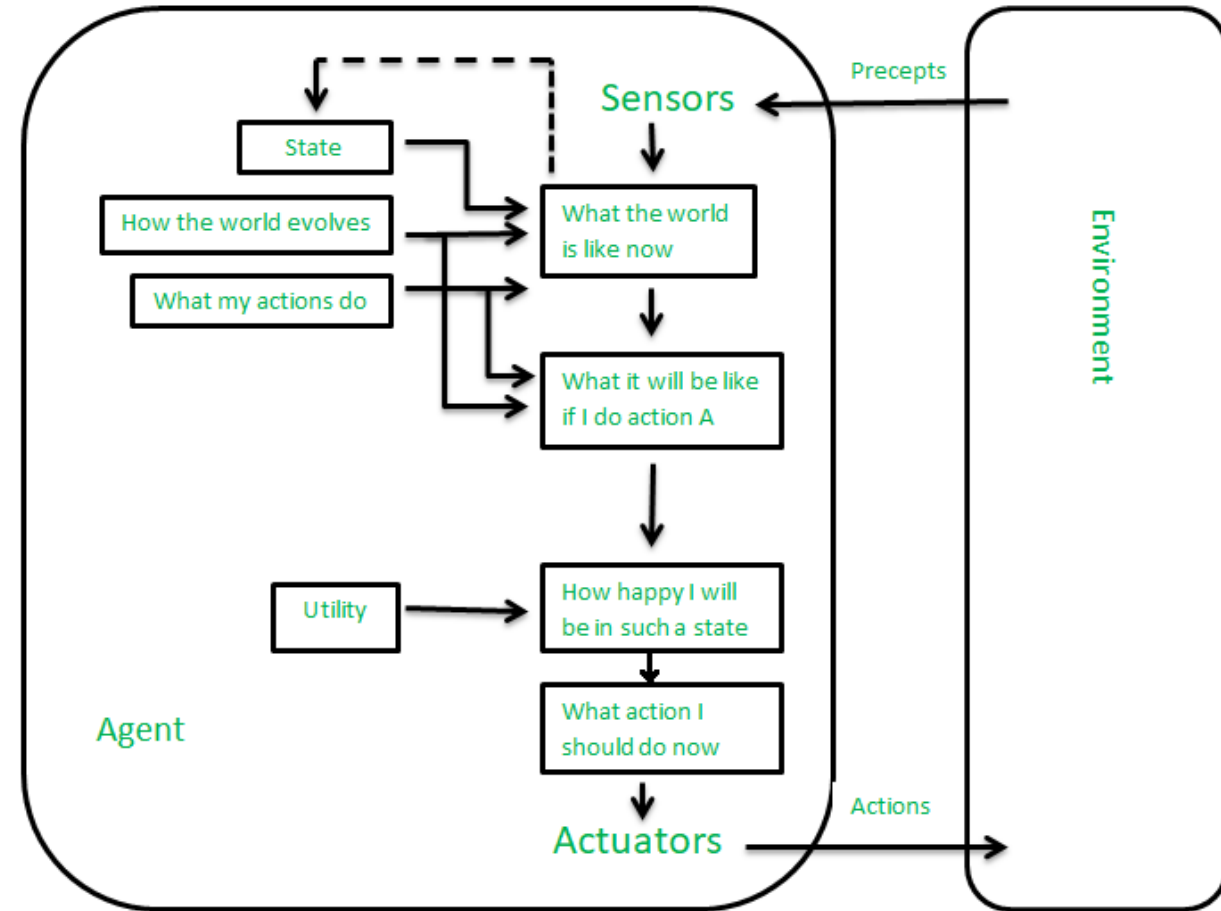


# Agent types

- Utility-based agents

## Tác tử dựa trên lợi ích

- Trong nhiều môi trường, thông tin về các mục tiêu không đủ để đánh giá hiệu quả của các hành động
  - Có rất nhiều chuỗi các hành động cho phép taxi đi đến đích (tức là đạt đến mục tiêu)
  - Nhưng: chuỗi hành động nào nhanh hơn, an toàn hơn, đáng tin cậy hơn, chi phí thấp hơn?
- Cần sự đánh giá lợi ích đối với tác tử
- Hàm lợi ích (utility function)
  - Ánh xạ từ **chuỗi** các trạng thái của môi trường tới một giá trị số thực (thể hiện mức lợi ích đối với tác tử)

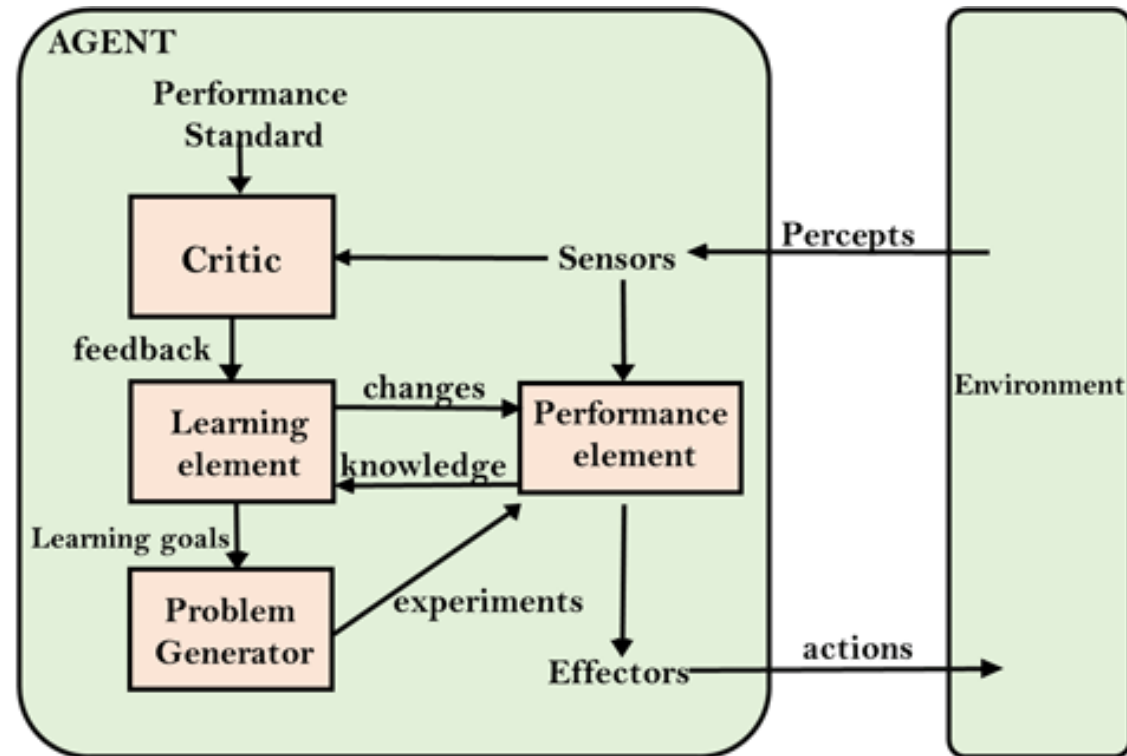


# Agent types

- Learning Agent

Tác tử có khả năng học

- Khả năng học cho phép tác tử cải thiện hiệu quả hoạt động của nó
- 4 thành phần tạo nên một tác tử có khả năng học
  - Thành phần *hành động*: đảm nhiệm việc lựa chọn các hành động
  - Thành phần đánh giá (bình luận): đánh giá hiệu quả hoạt động
  - Thành phần *học*: giúp cải thiện hiệu quả hoạt động - dựa trên các đánh giá, để thay đổi (cải thiện) thành phần hành động
  - Thành phần *sản sinh kinh nghiệm*: có nhiệm vụ đề xuất các hành động giúp sản sinh ra (dẫn đến) các kinh nghiệm mới





# Cơ sở tri thức của tác tử

---

- Tác tử cần có khả năng:
  - Thu thập (cập nhật) các tri thức mới
  - Cập nhật việc biểu diễn (bên trong tác tử) đối với môi trường xung quanh
  - Suy diễn ra các thuộc tính ẩn của môi trường xung quanh
  - Suy luận để đưa ra cá hành động hợp lý

**Thanks for your attention!**

**Q&A**

---