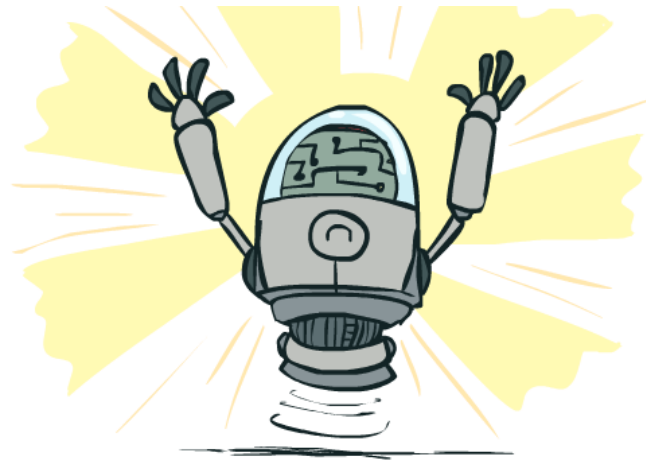




## 第二章 智能体

## Intelligent Agents



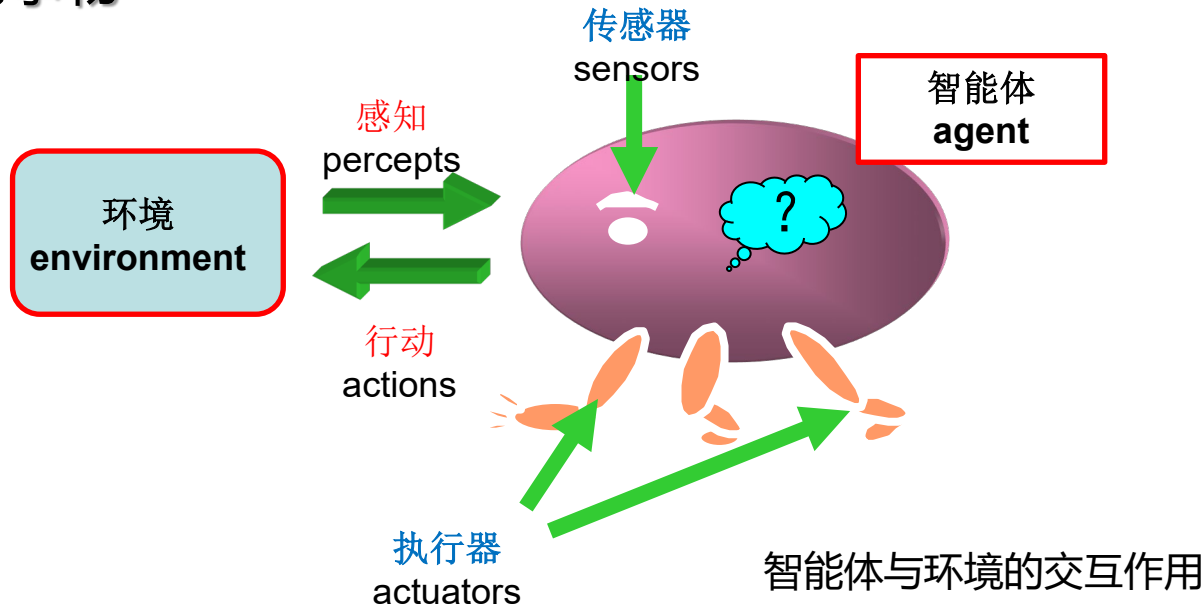
# Outline

- 2.1 智能体与环境
- 2.2 理性的概念
- 2.3 环境的性质PEAS
- 2.4 智能体的结构
- 2.5 小结



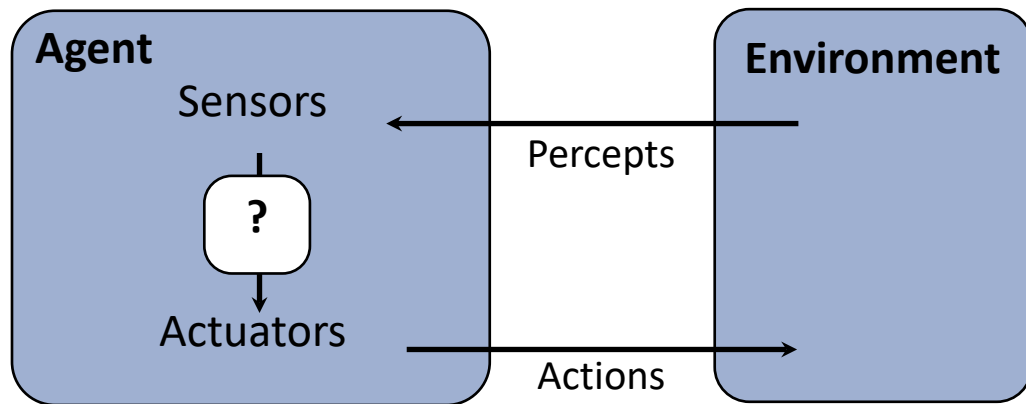
# 智能体 (Agent)

- **Agent的概念**：通过**传感器感知**所处环境，并通过**执行器**对该环境产生**作用**的事物



# 智能体 (Agent)

- **Agent**通过传感器感知环境，并通过执行器对所处的环境产生影响。
  - **感知**：表示任何给定时刻Agent的感知输入
  - **感知序列**：该Agent所收到的所有输入数据的完整历史



Agent在任何给定时刻的行动选择依赖于到那个时刻为止，该Agent感知到的序列

# 智能体举例

## 人类智能体

传感器：眼睛、耳朵等

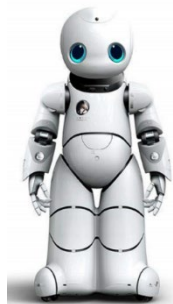
执行器：手、腿等



## 机器人智能体

传感器：摄像头、红外测距仪等

执行器：各种马达等



## 自动驾驶汽车智能体

传感器：摄像头、激光雷达等

执行器：加速器、制动器、方向盘等



# 智能体定义

- 智能体：给定感知序列 - - 求取行动（映射）

通过智能体函数来实现，抽象的数学表示：

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

智能体函数通过具体的智能体程序来实现，并在智能体自身的结构上运行

因此，智能体定义为：

智能体 = 体系结构 + 程序



# 智能体 - 体系结构

- **体系结构**：某个具备物理传感器和执行器的计算装置。
- 例如：一台计算机，一辆自动驾驶汽车



- 体系结构为程序**提供**来自传感器的**感知信息**，**运行程序**，并把程序计算出的**行动决策送达**执行器。

# 举例：真空吸尘器世界



现实世界open

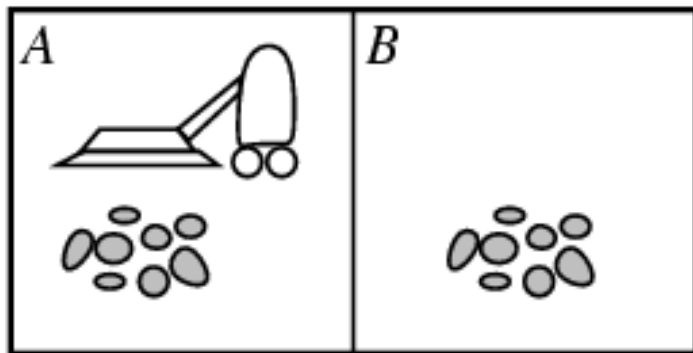
问题假设



真空吸尘器世界 closed



# 举例：真空吸尘器世界

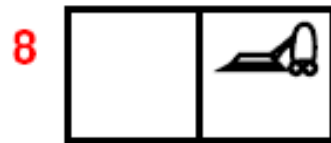
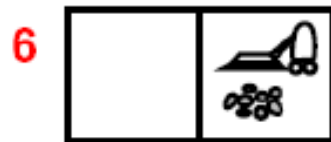
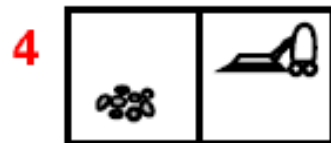
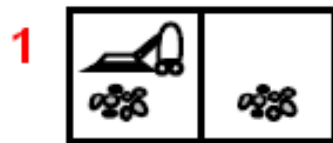


- **感知 (Percepts):** 位置和内容, 如[A, Dirty]
- **行动 (Actions):** 向左移动、 向右移动、 吸灰尘、 什么都不做

# 举例：真空吸尘器世界

- Simple world

- 8 States (状态)



# Vacuum-cleaner world

尝试定义一个**Agent函数**:

如果当前有灰尘, 那么吸尘, 否则移动到另一处地点

真空吸尘器世界的**简单Agent函数**的部分列表

| 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   |
| ...                    | ...    |

填写表格右边  
一列的方法不  
同, 就可以定  
义不同的真空  
吸尘器世界  
**Agent**。

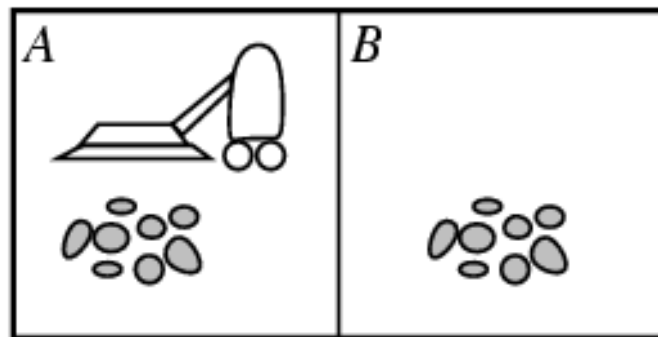
# Table-lookup agent

- **表驱动法**：一个表记录所有的感知序列及其最佳行动。在行动时，只需要根据感知，查询表格来决定行动
- **缺陷**：
  - 表太大
  - 建表要花很长时间
  - 不是自主的
  - 即使是**学习**，也要花很长时间学习表中的条目
  - ...
- **举例**（真空吸尘器、国际下棋、自动驾驶）

# Vacuum-cleaner world

**Agent函数:**

如果当前有灰尘,  
那么吸尘, 否则移动到  
另一处地点



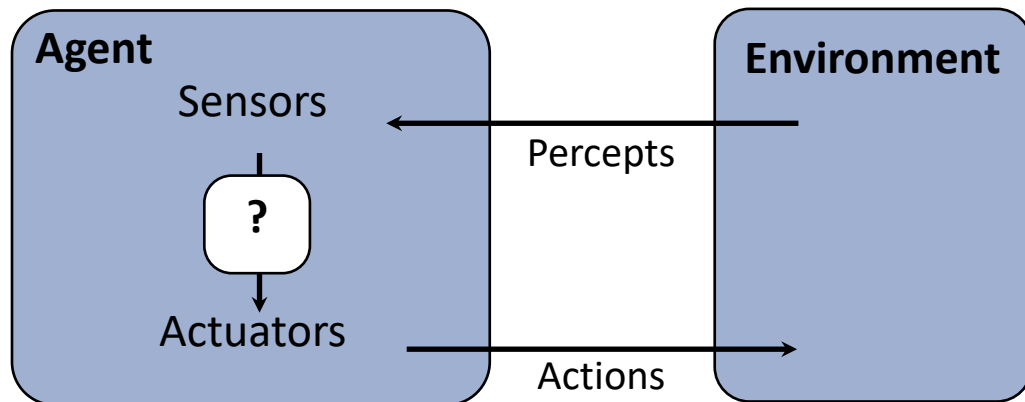
**Agent程序**是Agent函数的实现

```
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
```



# 举例：真空吸尘器世界

**function 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*



智能体程序

# Outline

---

- 2.1 智能体与环境
- 2.2 理性的概念
- 2.3 环境的性质PEAS
- 2.4 智能体的结构
- 2.5 小结

# 理性智能体 (Rational agents)

---

- 初步理解：做正确事情的智能体（基于所感知的，采取“对”的行动，使智能体更加成功）
- 性能度量：衡量智能体成功程度的一个客观标准，例如：
  - 真空吸尘器世界
    - 清理灰尘的数量、所花时间
    - 耗电量、产生的噪声.....



# Rational agents

- 任何指定时刻，根据已知的**感知序列**和智能体具有的**先验知识**，理性智能体应该选择期望能使其**性能度量最大化**的行动。
- 理性的判断依赖于以下4个方面：
  - 定义成功标准的**性能度量**
  - Agent对环境的**先验知识**
  - Agent可以采取的**行动**
  - Agent截止到当前时刻的**感知序列**

# 理性与全知（omniscience）的区别

- 理性 ≠ 全知

- 一个全知的智能体明确的知道它的行动产生的实际结果并且做出相应的动作，全知者在现实中是不可能的。
- 理性并不要求全知。理性的选择只依赖于到当时为止的感知序列。

- 理性 ≠ 完美

- 理性是使期望的性能最大化
- 完美是使实际的性能最大化
- 对智能体而言，完美是不太合理的要求

# Outline

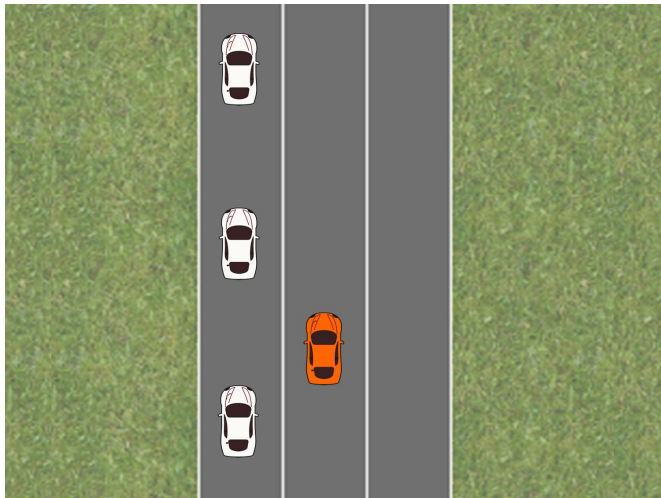
---

- 2.1 智能体与环境
- 2.2 理性的概念
- 2.3 环境的性质PEAS
- 2.4 智能体的结构
- 2.5 小结

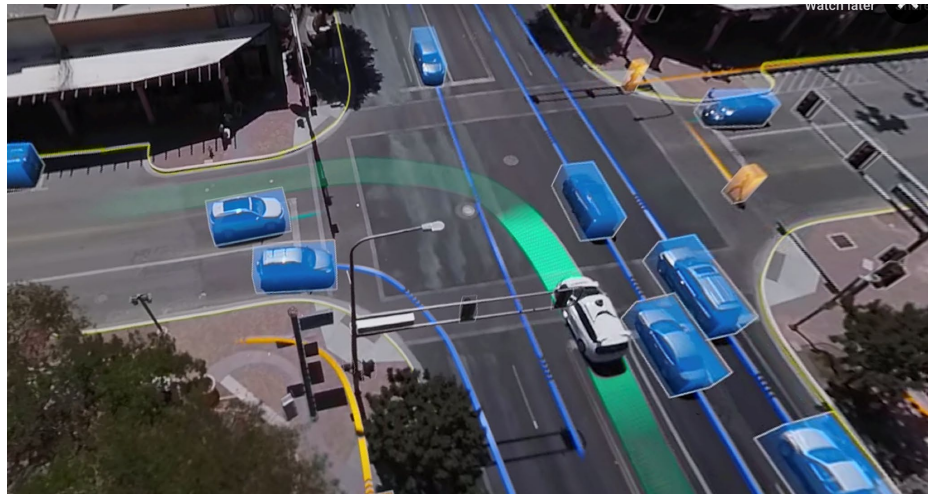
## 2.3.1 任务环境

- 设计智能体第一步

尽可能全面详细地说明任务环境



Drive safely along the highway?



Drive safely along the Cao'an Road?

## 2.3.1 任务环境

- 设计智能体第一步

尽可能全面详细地说明任务环境



Win an art competition?



Write an intentionally funny story?



You

Please write me a very funny, and extremely short story.

< 6 / 6 >



ChatGPT

Once there was a chicken who was terribly afraid of crossing roads. One day, it mustered up all its courage, stepped onto the road, and halfway across, it suddenly stopped and said, "Wait, why am I doing this again?"

< 12 / 12 >    

## 2.3.1 任务环境

---

- 设计智能体第一步

尽可能全面详细地说明任务环境

- PEAS描述
  - 性能度量 (Performance measure)
  - 环境 (Environment)
  - 执行器 (Actuators)
  - 传感器 (Sensors)

# 任务环境（续）

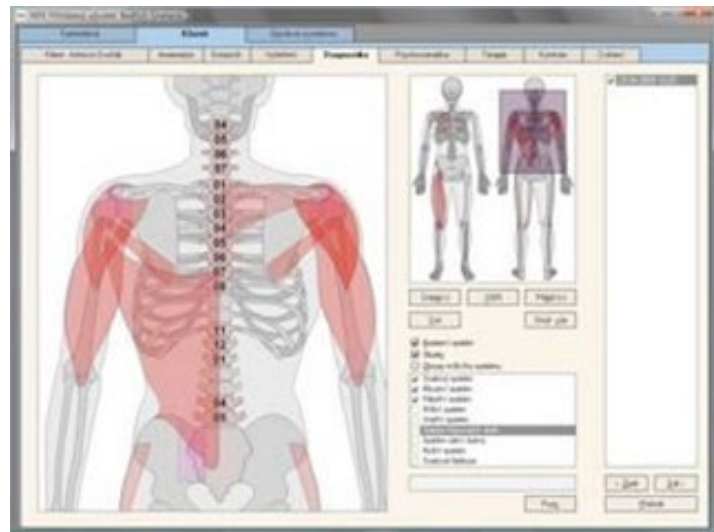
- 挑拣零件的机器人
  - **性能度量**: 放进正确的箱子的零件数
  - **环境**: 载有零件的传送带、箱子
  - **执行器**: 有关节的胳膊和手
  - **传感器**: 摄像头、关节



# 任务环境（续）

- 医学诊断系统

- **性能度量**: 病人恢复健康、费用最小、最小诉讼等
- **环境**: 病人、医院、医生等
- **执行器**: 显示器(问题、测试、诊断、资料、咨询等)
- **传感器**: 键盘(症状、病人的回答、检查报告等)





# 任务环境（续）

- 自动出租车

- **性能度量:** 安全、快速、守法、舒适旅途、利润最大化等
- **环境:** 道路、其他车辆、行人、顾客等
- **执行器:** 方向盘、加速器、刹车、信号灯、喇叭、显示器等
- **传感器:** 摄像头、声波传感器、速度计、GPS、里程计、加速计、引擎传感器、键盘等



## 2.3.2 环境类型

- 环境的类型很大程度上决定了智能体的设计

|         | 计时棋赛      | 不计时棋赛     | 自动出租车驾驶 |
|---------|-----------|-----------|---------|
| 完全可以观察的 | Yes       | Yes       | No      |
| 确定性的    | Strategic | Strategic | No      |
| 静态的     | Semi      | Yes       | No      |
| 离散的     | Yes       | Yes       | No      |
| 单个Agent | No        | No        | No      |

## 2.3.2 环境类型（续）

- 环境的类型很大程度上决定了智能体的设计，最难处理的情况(现实世界)，以自动驾驶为例：
  - **部分可观测**=>Agent不能在每个时间节点上都能获取环境的**完整状态**
  - **随机的**=> Agent不得不应对一些**突发情况**
  - **多Agent**=>部分合作部分竞争多Agent环境
  - **动态的**=>Agent决策时环境会**随着时间变化**
  - **连续的**=>连续地执行一系列的**控制行动**



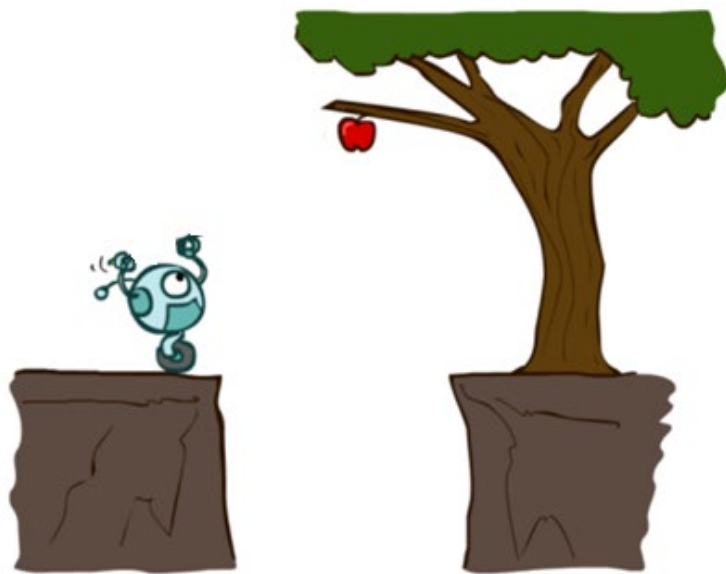
# Outline

---

- 2.1 智能体与环境
- 2.2 理性的概念
- 2.3 环境的性质PEAS
- 2.4 智能体的类型
- 2.5 小结

# 智能体类型

- 简单反射Agent
- 基于模型的反射Agent
- 基于目标的Agent
- 基于效用的Agent
- 所有Agent都可以通过学习来改进性能



# 简单反射Agent

- Simple reflex agent is the simplest type of Agent
- “世界**现在**怎么样了？”
- 这类Agent基于**当前的感知选择行动**，不关注感知历史

```
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
```

# Simple reflex agents

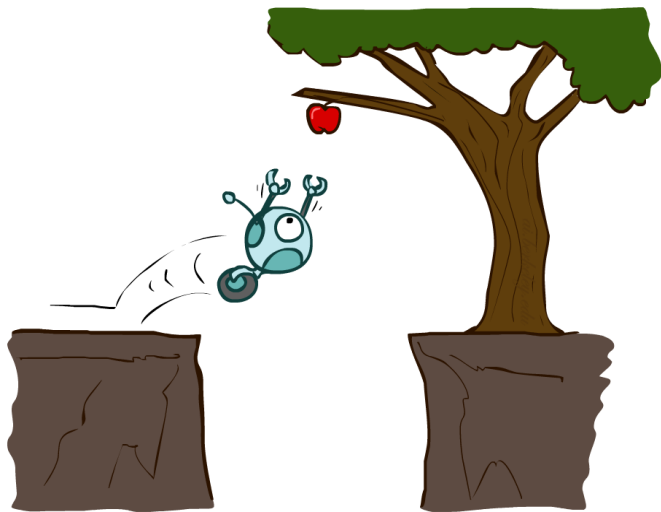
- 背景知识：条件-行为规则
  - 如果有灰尘，那么吸尘
  - 如果前方车辆刹车，那么刹车

人类也有很多这样的联接，

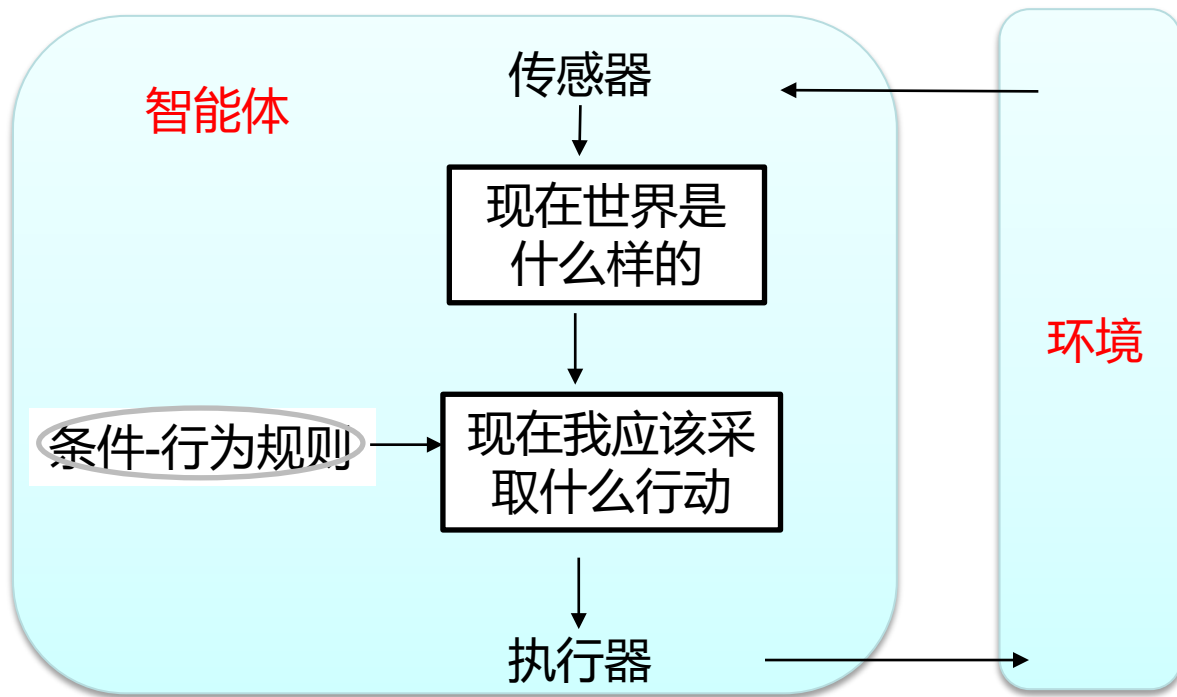
有些是学习得到的反应

（驾车），有些是先天反

射（眨眼）



# 简单反射型智能体



例如：真空吸尘器

- 矩形表示Agent决策过程的内部状态
- 椭圆表示该过程中用到的背景知识



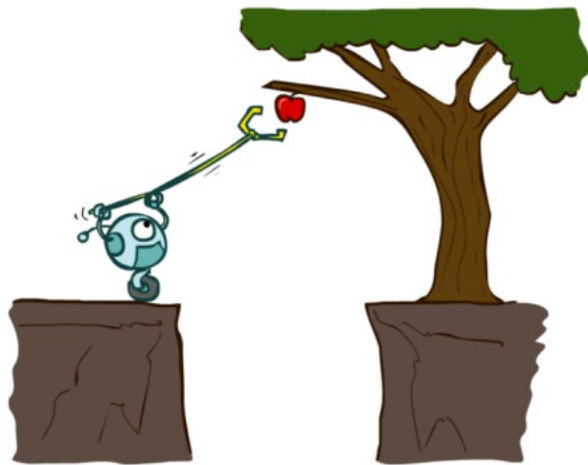
# 基于模型的反射型智能体

- 处理部分可观测环境的有效途径，是让Agent跟踪记录历史感知（现在看不到那部分世界）
- Agent应该根据感知历史，维持内部状态

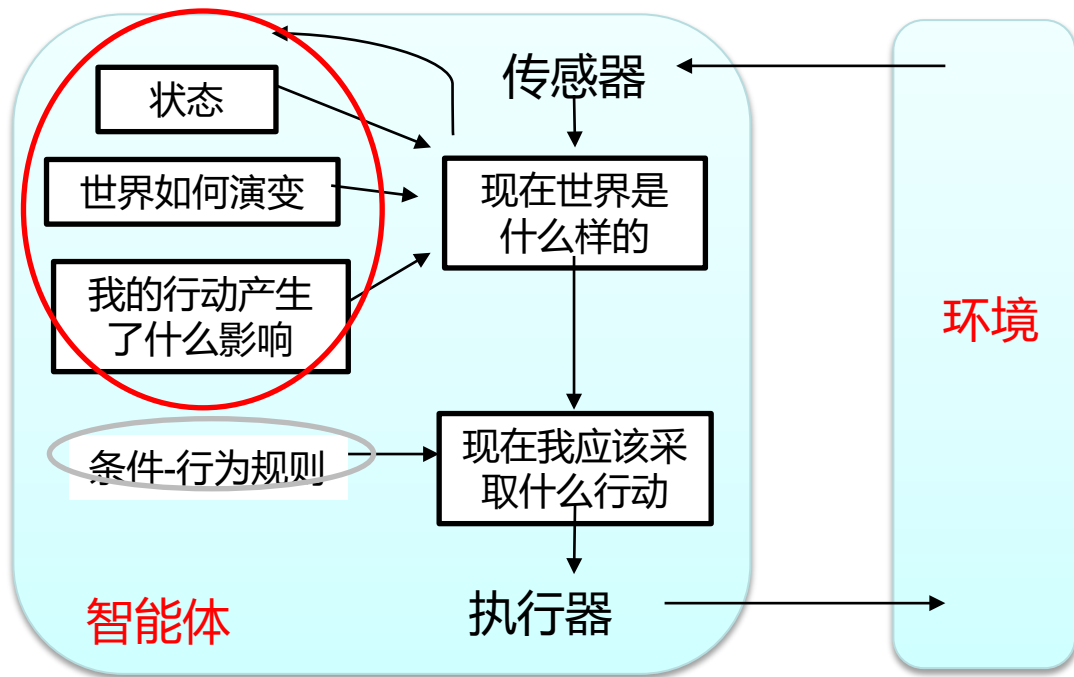
要求Agent程序中加入两类知识：

世界是如何独立于Agent而发展的信息

Agent自身的行动如何影响世界的信息



# 基于模型的反射型智能体



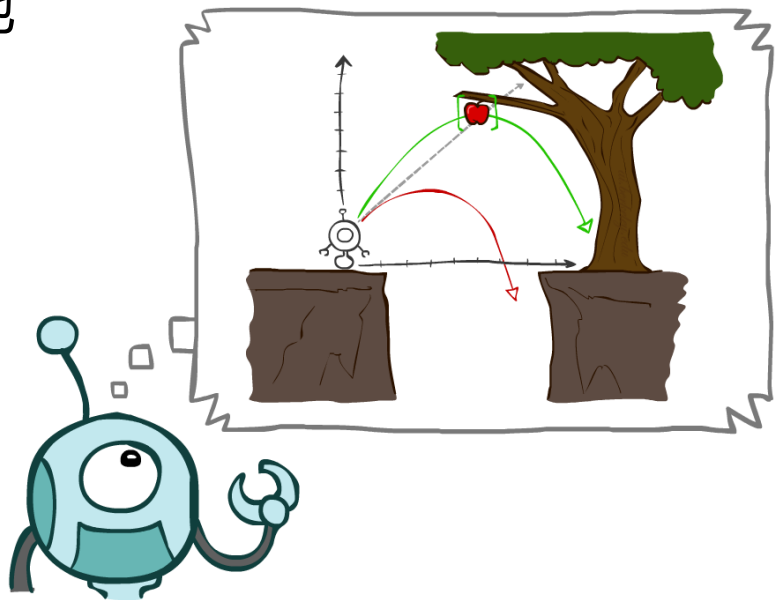
世界模型：1) 世界如何运转的？2) 之前采取了什么样的行动？

利用当前感知的信息与过去的内部状态结合起来更新当前状态。

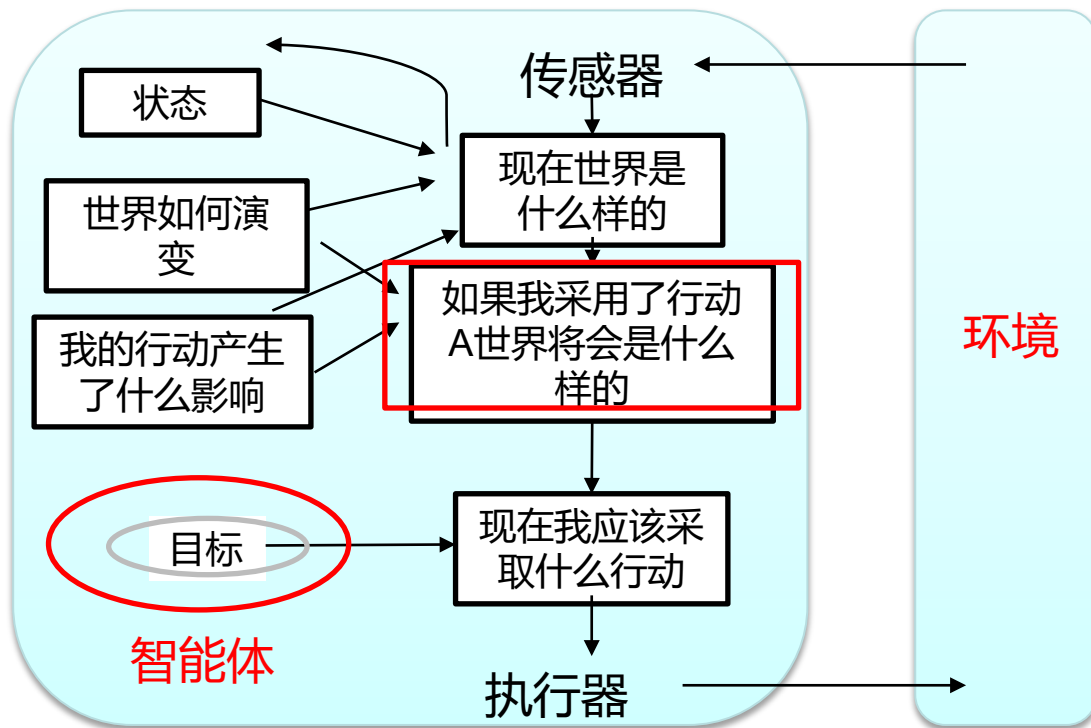
背景知识：rules、state、model、action

# 基于目标的智能体

- 除了当前状态和历史经验，还需要**目标信息**来描述想要达到的状况
  - 出租车：乘客要到达的目的地
- 与反射Agent的根本不同：
  - 反射Agent内建的规则直接把感知映射到了行动
  - 目标Agent则考虑了未来的影响，原则上要求会推理



# 基于目标的智能体

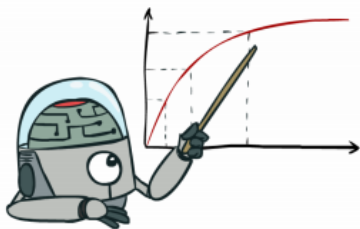


# 基于效用的智能体

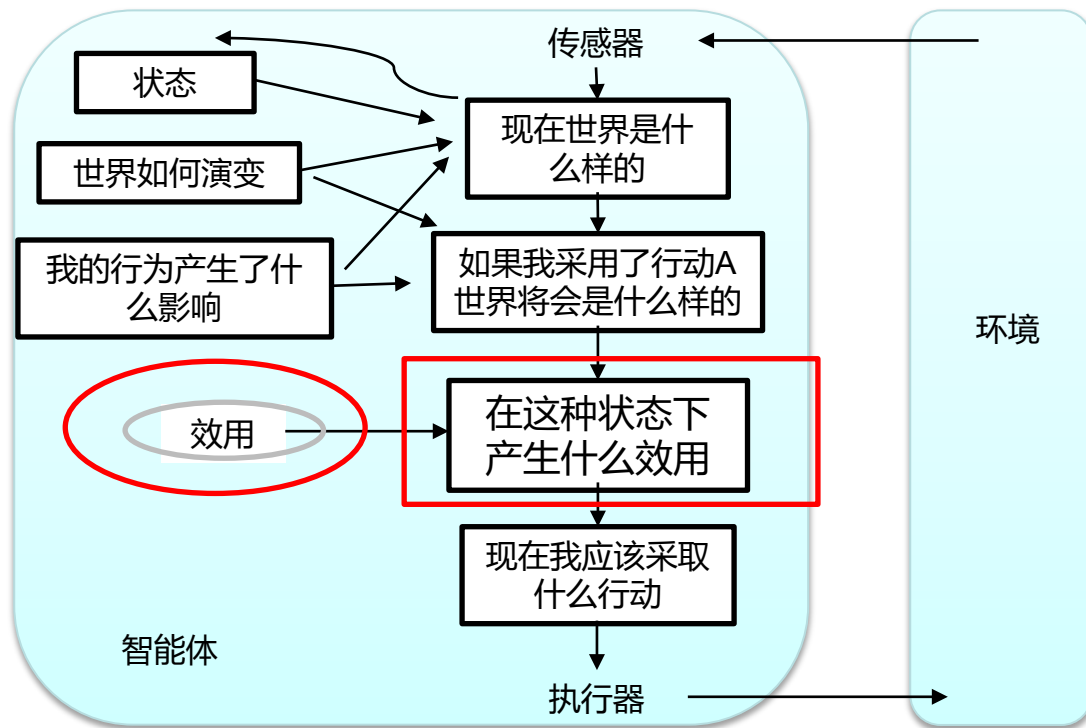
- 目标在很多环境中不足以生成高品质的行为
  - 出租车：到达目的地
  - 更快、更安全、更可靠或者更便宜

效用函数是性能度量的内在化，如果内在的效用函数和外在的性能度量是和谐的，那么选择最大效用行动的Agent根据外在的性能度量也是理性的。

## Maximize Your Expected Utility



# 基于效用的智能体



使用了世界模型和度量对各个状态偏好程度的效用函数

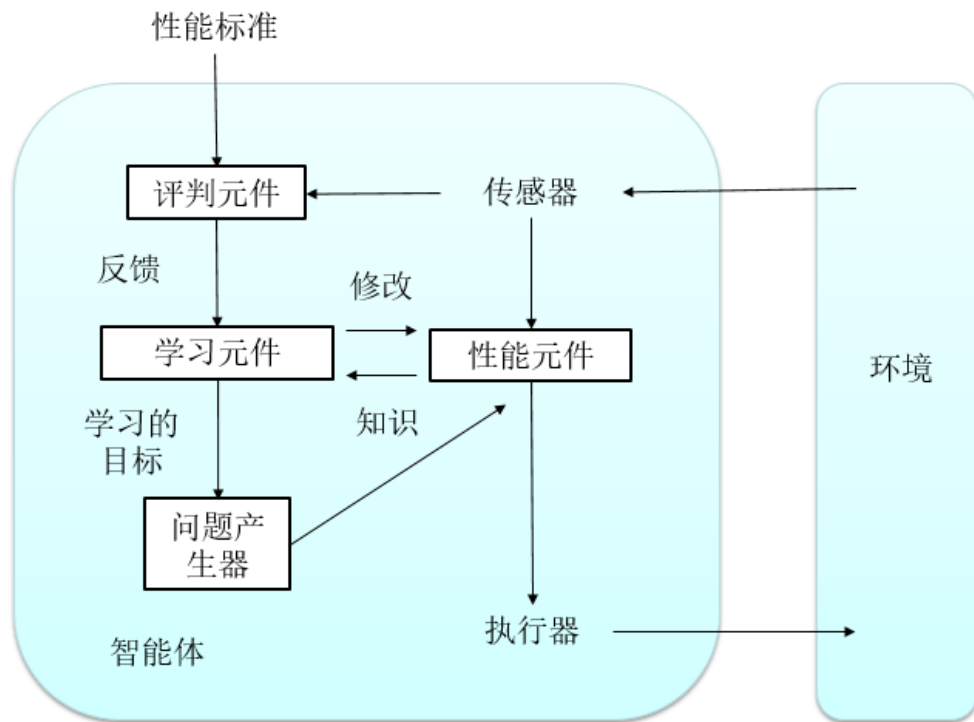
# Learning agents

---

- 1950年，图灵一篇里程碑式的论文，机器能思考吗？为人类带来了一个新学科—人工智能。
- 提出人工编程序实现智能机器的思想；估计了这可能需要的工作量，结论是“看来需要某种更迅速的方法”
- 提出了建造会学习的机器，然后教育它们。

# Learning agents

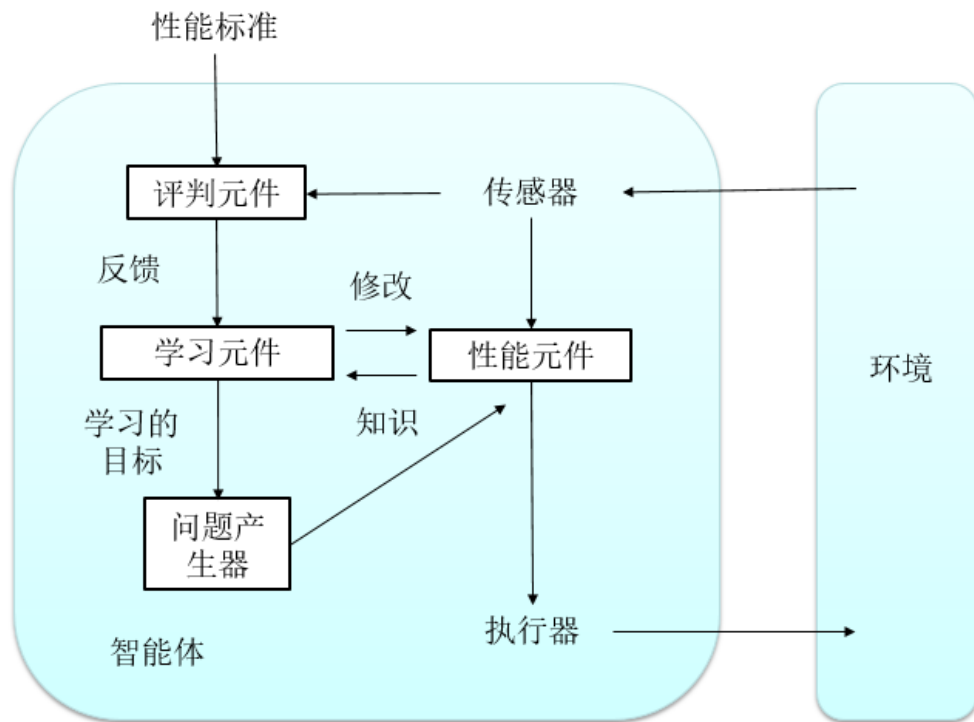
- 学习Agent的四个组件：
  - **性能元件**
    - Agent,接受感知信息并决策
  - **学习元件**
    - 修改性能元件使其在未来做的更好
  - **评判元件**
    - 观察世界并把信息传递给学习元件
  - **问题产生器**
    - 建议少量的探索行为





# Learning agents

- 以无人驾驶为例：
  - **性能元件**: 选择驾驶行动
  - **评判元件**: 快速左转横穿三条车道的后果
  - **学习元件**: 根据经验修改性能元件
  - **问题产生器**: 提议不同路面的刹车实验



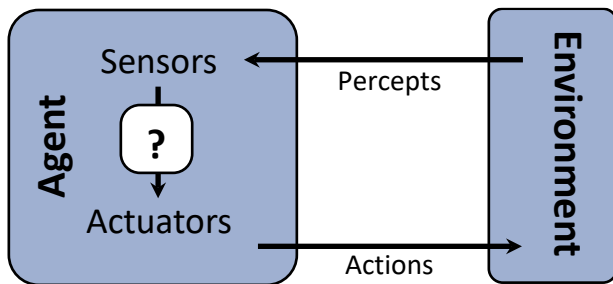
# Outline

---

- 2.1 智能体与环境
- 2.2 理性的概念
- 2.3 环境的性质PEAS
- 2.4 智能体的结构
- 2.5 小结

# AI as Designing Rational Agents

- An **agent** is an entity that *perceives* and *acts*.
- A **rational agent** selects actions that maximize its expected utility.



## 2.5 小结

本章旋风般地介绍了人工智能，我们将之称为 Agent 设计的科学。要点回顾如下：

- **Agent** 是可以感知环境并在环境中行动的事物。**Agent** 函数指定 Agent 响应任何感知序列所采取的行动。
- 性能度量评价 Agent 在环境中的行为表现。给定 Agent 的感知序列，**理性 Agent** 行动追求性能度量预期值最大化。
- **任务环境**的规范包括性能度量、外部环境、执行器和传感器。设计 Agent 时，第一步总是把任务空间定义得尽可能完全。
- 任务环境从不同的维度看有很多变化。它们可能是完全或部分可观察的，单 Agent 或多 Agent 的，确定性的或随机的，片段式的或延续式的，静态的或动态的，离散的或连续的，已知的和未知的。
- **Agent 程序**是 Agent 函数的实现。有各种基本的 Agent 程序的设计，反映出显式表现的以及用于决策过程的信息种类。设计可能在效率、压缩性和灵活性方面有变化。适当的 Agent 程序的设计依赖于环境的本性。
- **简单反射 Agent** 直接对感知信息做出反应，**基于模型的反射 Agent** 保持内部状态，追踪记录当前感知信息中反映不出来的世界各方面。**基于目标的 Agent** 的行动是为了达到目标，而**基于效用的 Agent** 试图最大化它期望的“快乐”。
- 所有 Agent 都可以通过学习来改进它们性能。



---

THANKS!

---