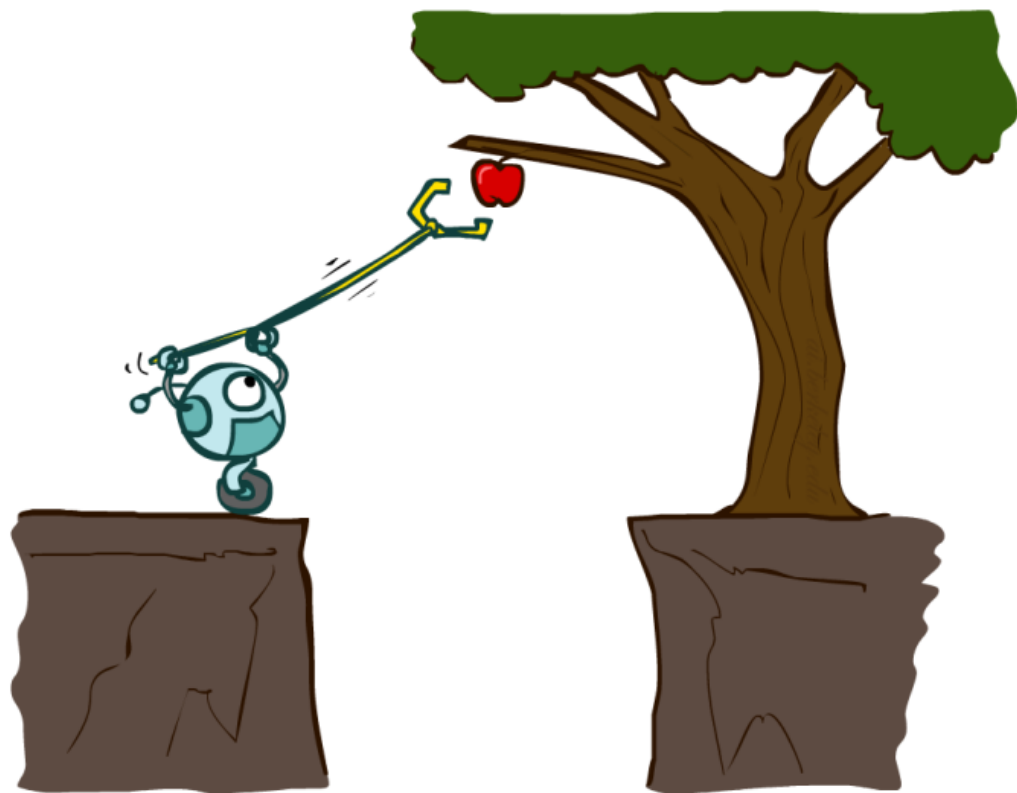


# 人工智能： 智能体和 环境

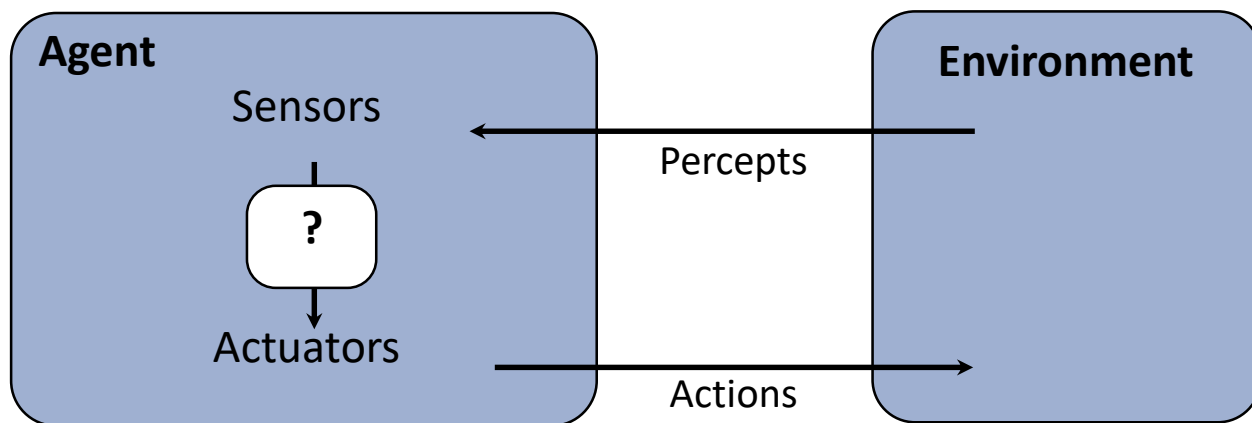
---



## 提纲

- 智能体和环境
- 合理性
- PEAS (性能指标, 环境, 促动器, 传感器)
- 环境类型
- 智能体类型

# 智能体和环境



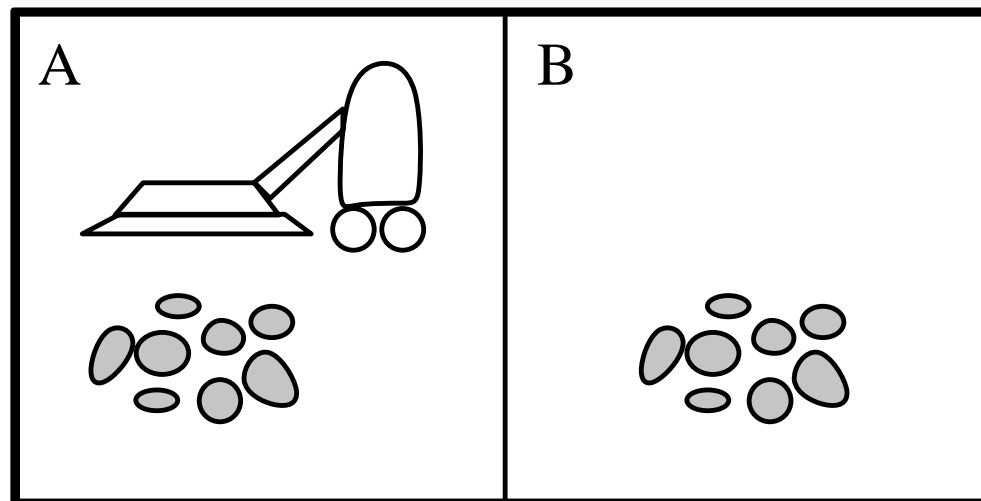
# 智能体和环境

- 一个智能体通过自身的**传感器**感知环境，并通过自身的**促动器**去相应的**行动**。
- 人是智能体
  - 传感器 = 视觉，听觉，触摸，嗅觉，味觉，主体感觉
  - 促动器（激励器） = 肌肉，分泌物质，大脑状态的改变
- 口袋计算器
  - 传感器 = 按键状态传感器
  - 促动器 = 数字显示
- 人工智能主要研究对象
  - 有自己的计算资源和能力的智能体
  - 和所处环境要求相对复杂的决策过程

# 智能体函数和智能体程序(agent functions and agent programs)

- 智能体函数: 感知历史 (序列) 到行动的映射
  - $f: \mathcal{P}^* \rightarrow \mathcal{A}$
- 智能体程序  $l$ , 运行在某个计算机  $M$ , 以实现  $f$ :
  - $f = \text{Agent}(l, M)$
  - 运行在智能体内
  - $M$ , 有限的运算和存储空间
  - 程序运行可能比较慢; 也可能被新的感知所打断 (或忽略它们)。
  - 不是每一个智能体函数都能被实现
    - $M$  的限制

# 举例：吸尘器

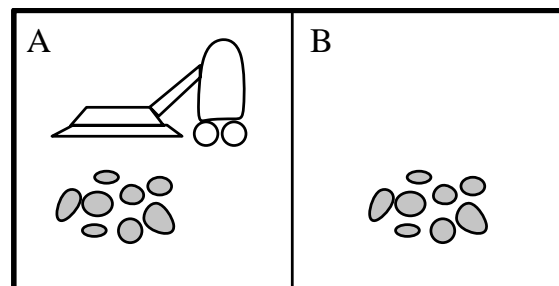


- 感知对象：[地点， 洁净状态] ([A, 有灰尘])
- 行动：左， 右， 吸尘， 无动作

# 吸尘器智能体

## 智能体函数

| 感知序列              | 行动   |
|-------------------|------|
| 【A, 干净】           | 向右移动 |
| 【A, 灰尘】           | 吸尘   |
| 【B, 干净】           | 向左移动 |
| 【B, 灰尘】           | 吸尘   |
| 【A, 干净】 , 【B, 干净】 | 向左移动 |
| 【A, 干净】 , 【B, 灰尘】 | 吸尘   |
| ...               | ...  |



## 智能体程序

**Function** 简单吸尘器([位置,状态]) **returns** 一个行动

**If** 状态 = 灰尘 **then return** 吸尘

**else if** 位置 = A **then return** 向右移动

**else if** 位置 = B **then return** 向左移动

# 合理性 (Rationality)

固定的性能指标  
(performance measure) 衡量  
环境状况

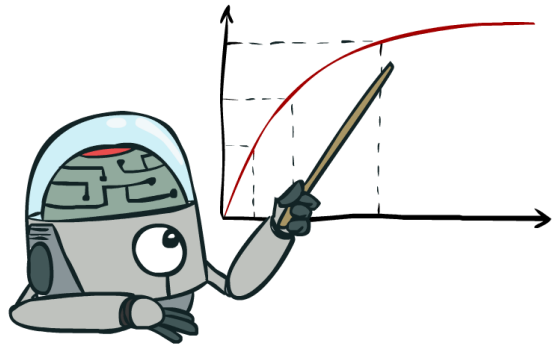
- 评估智能体的行为所导致环境变化后的状态
- 比如，每小时每一平米干净的地面加一分
- 关注的是环境状态，而不是智能体的态度、状态
- 同样的性能指标，可能被不同行为的智能体所成就；所带来的思考

一个合理的智能体 (rational agent)

- 给定最新的感知序列，和关于环境的预先知识(prior knowledge)，选择为了最大化性能指标期望值的行动。
- 问题：之前定义的吸尘器智能体实现的是一个合理的智能体函数吗？



## 合理性，继续



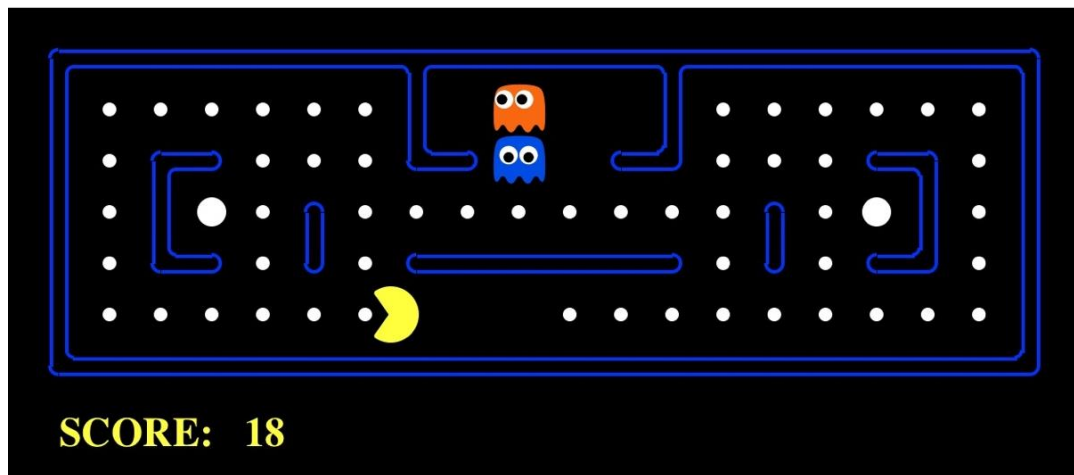
- 合理性决定于4个因素
  - 性能指标
  - 智能体的预先知识
  - 智能体能够采取的行动集合
  - 智能体的感知序列
- 成就合理性意味着 **最大化你的期值**

# 合理性，继续

- 合理的智能体对环境是无所不知的吗？
  - 不是。受限于传感器及其感知。
- 合理的智能体具有超常的洞察力吗？
  - 不是。可能缺少对环境动态变化的掌握。
- 那么，合理的智能体在环境中探索和学习吗？
  - 是的。只有具备这些，才能在未知的环境中行动。
- 合理的智能体不尽然都会表现的成功，但是，
  - 它们都是有自主性的（不是简单规则化的选择行为动作，超越初始的程序）

# 任务环境描述-- PEAS

- 性能指标(Performance measure)
  - -1 每走一步; +10 一个豆子;  
+500 赢; -500 输...
- 环境 (Environment)
  - 整个游戏环境, pacman和妖怪的动态行为
- 促动器(Actuators)
  - 左、右、上、下
- 传感器 (Sensors)
  - 整个游戏环境状态可见



# 环境描述： 自动驾驶出租车

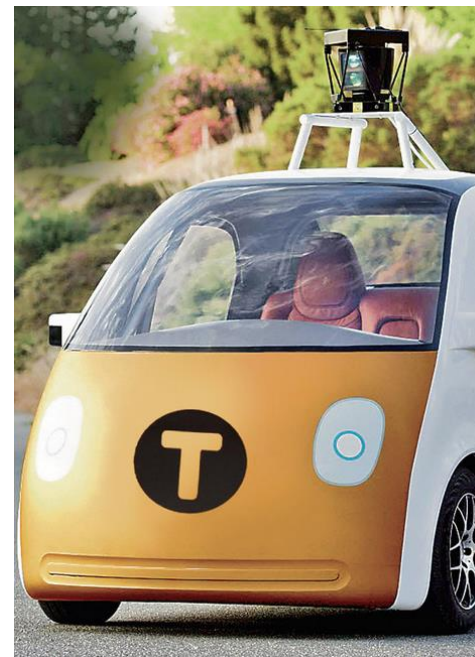
- 性能指标
- 环境
- 促动器
- 传感器



Image: <http://nypost.com/2014/06/21/how-google-might-put-taxi-drivers-out-of-business/>

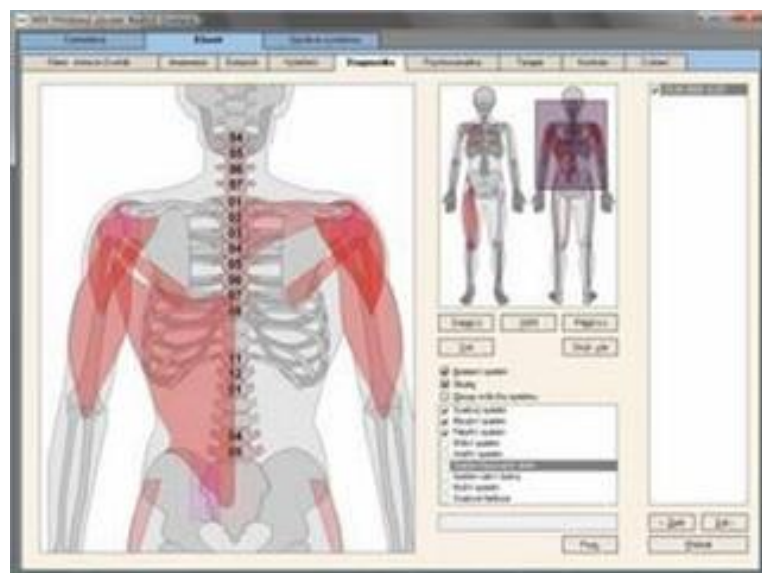
# 环境描述： 自动驾驶出租车

- 性能指标
  - 收入， 乘客满意度， 油耗， 罚单， 保险费
- 环境
  - 街道， 乘客， 行人， 其他车辆
- 促动器
  - 方向盘， 刹车系统， 油门， 显示器/扬声器
- 传感器
  - 照相机， 雷达， 引擎传感器， 麦克风， 速度仪表



# 环境描述： 医疗诊断系统

- 性能指标
  - 患者康复, 费用, 诊断准确性
- 环境
  - 患者, 医护人员, 保险公司, 法院
- 促动器
  - 屏幕显示, 电子邮件
- 传感器
  - 键盘, 鼠标



# 环境类型 (属性)

全部可观察 vs  
部分可观察(fully  
observable vs  
partially  
observable)

- 每时每刻观察到全盘环境

单一智能体 vs  
多智能体 (single  
vs multi-agents)

- B的行为是最大化一个性能指标, 其数值由A的行为决定
- 竞争; 合作

变化确定性的 vs  
随机变化 (变化  
不确定) 的  
(deterministic  
vs stochastic)

- 当前状态和智能体行为决定环境下一个状态
- 忽略其他智能体行为所引发的不确定性
- 不确定(uncertain)环境: 在部分可观察, 或随机变化的环境

# 环境类型 (属性) 继续

由片段组成的  
vs 关联序列的  
(episodic vs  
sequential)

- 作业环境由独立的片段组成；每片段中，智能体观察一次，做出一个动作；每片段中的行为相互独立
- 序列环境中，智能体行为有相关性

静态 vs 动态的  
(static vs  
dynamic)

- 智能体在思考时，环境是否在改变
- 出租车自动驾驶；填字拼成语；计时象棋比赛

离散性 vs 连续性  
(discrete  
vs continuous)

- 决定于环境的状态，时间的处理，智能体的感知和行动
- 象棋，出租车自动驾驶



# 环境类型

| 环境属性     | Pacman | 医疗诊断 | 出租车 | 计时象棋 | 质检机器人 |
|----------|--------|------|-----|------|-------|
| 可观察性     |        |      |     |      |       |
| 智能体      |        |      |     |      |       |
| 环境变化确定性  |        |      |     |      |       |
| 环境变化的关联性 |        |      |     |      |       |
| 环境变化的动态性 |        |      |     |      |       |
| 状态连续性    |        |      |     |      |       |

# 环境类型

| 环境属性     | Pacman | 医疗诊断 | 出租车 | 质检机器人 |
|----------|--------|------|-----|-------|
| 可观察性     | 全部     | 部分   | 部分  | 部分    |
| 智能体      | 单个     | 单个   | 多个  | 单个    |
| 环境变化确定性  | 不确定    | 不确定  | 不确定 | 不确定   |
| 环境变化的关联性 | 连续的    | 连续的  | 连续的 | 片段的   |
| 环境变化的动态性 | 动态     | 动态   | 动态  | 动态    |
| 状态连续性    | 连续的    | 连续的  | 连续的 | 连续的   |

部分答案也依赖于环境是如何定义的。

# 智能体的设计

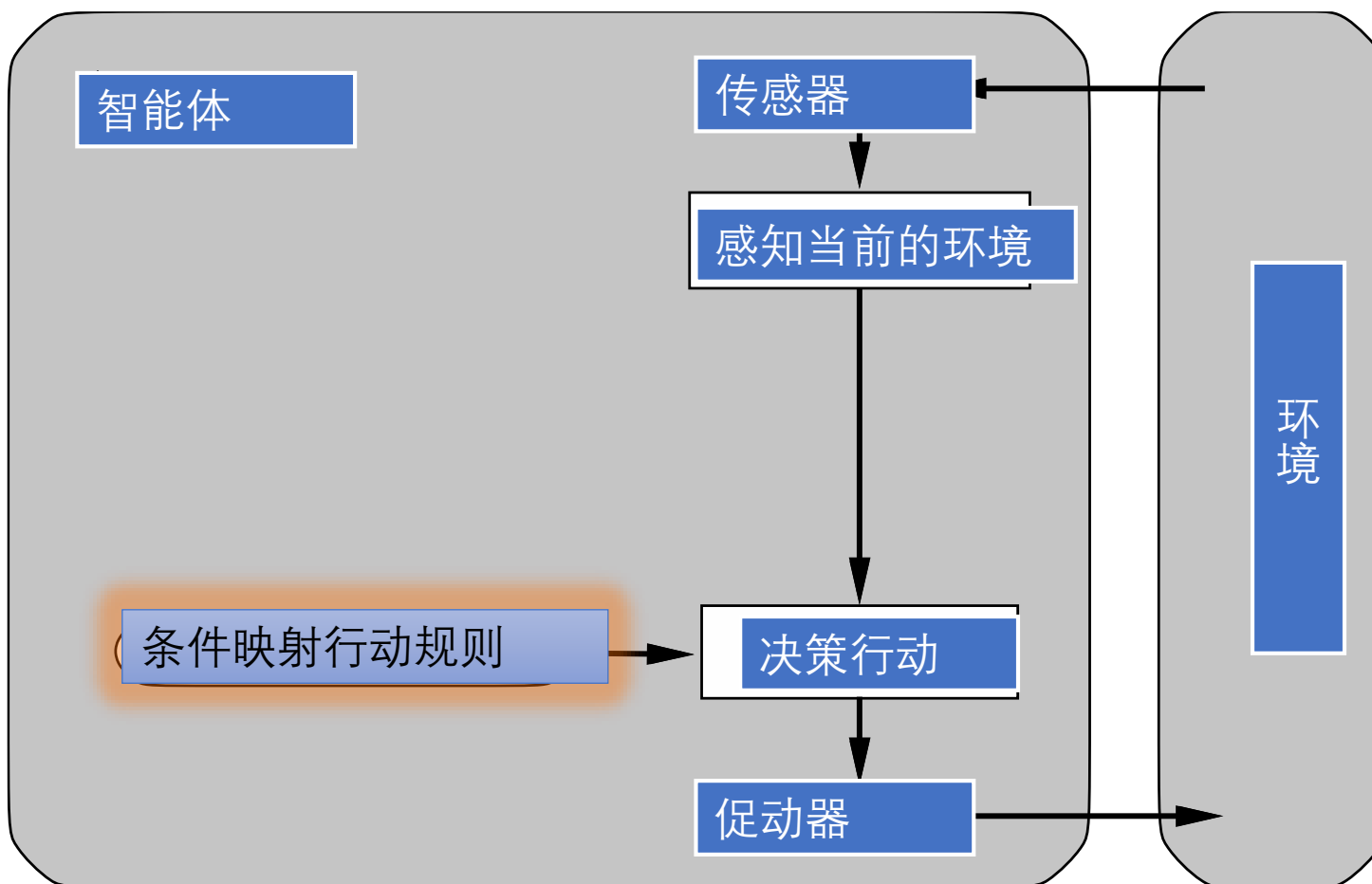
- 环境类型决定智能体的设计

- 这里不讨论其物理框架（包括计算机器，传感器，促动器）；只关于智能体程序的设计
- 部分可观察的 => 智能体需要存储内存
- 环境变化不确定的 => 需要准备偶发事件
- 多个智能体 => 可能需要一些随机的行为
- 静态的 => 有时间去计算一个合理决策
- 连续时间的 => 需要连续运行的控制器

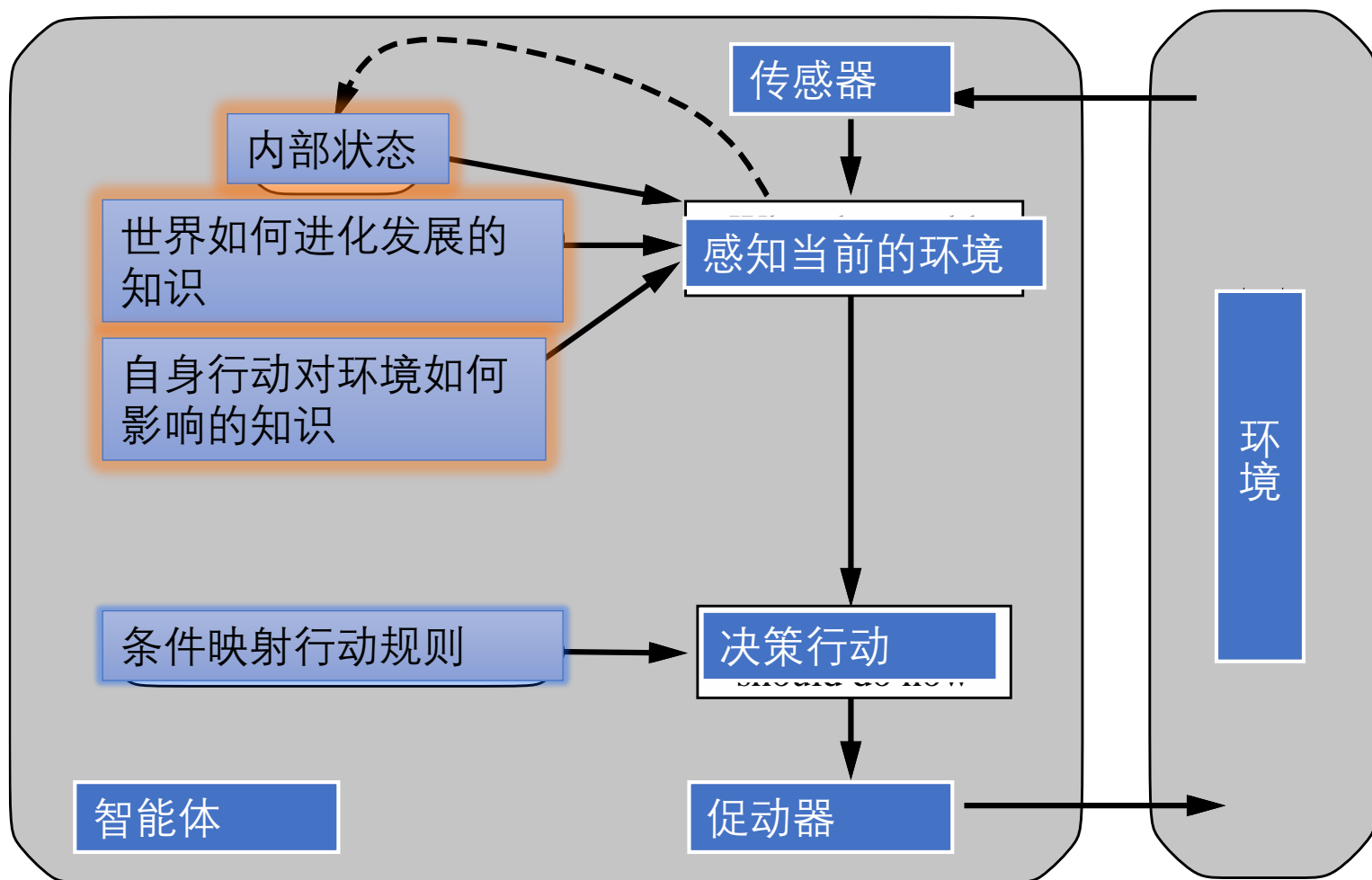
# 智能体类型

- 从简单的到复杂的
  - 简单反射型智能体(reflex)
  - 记录状态的反射智能体 (reflex with state)
  - 基于目标的智能体 (goal-based)
  - 基于功效的智能体 (utility-based)

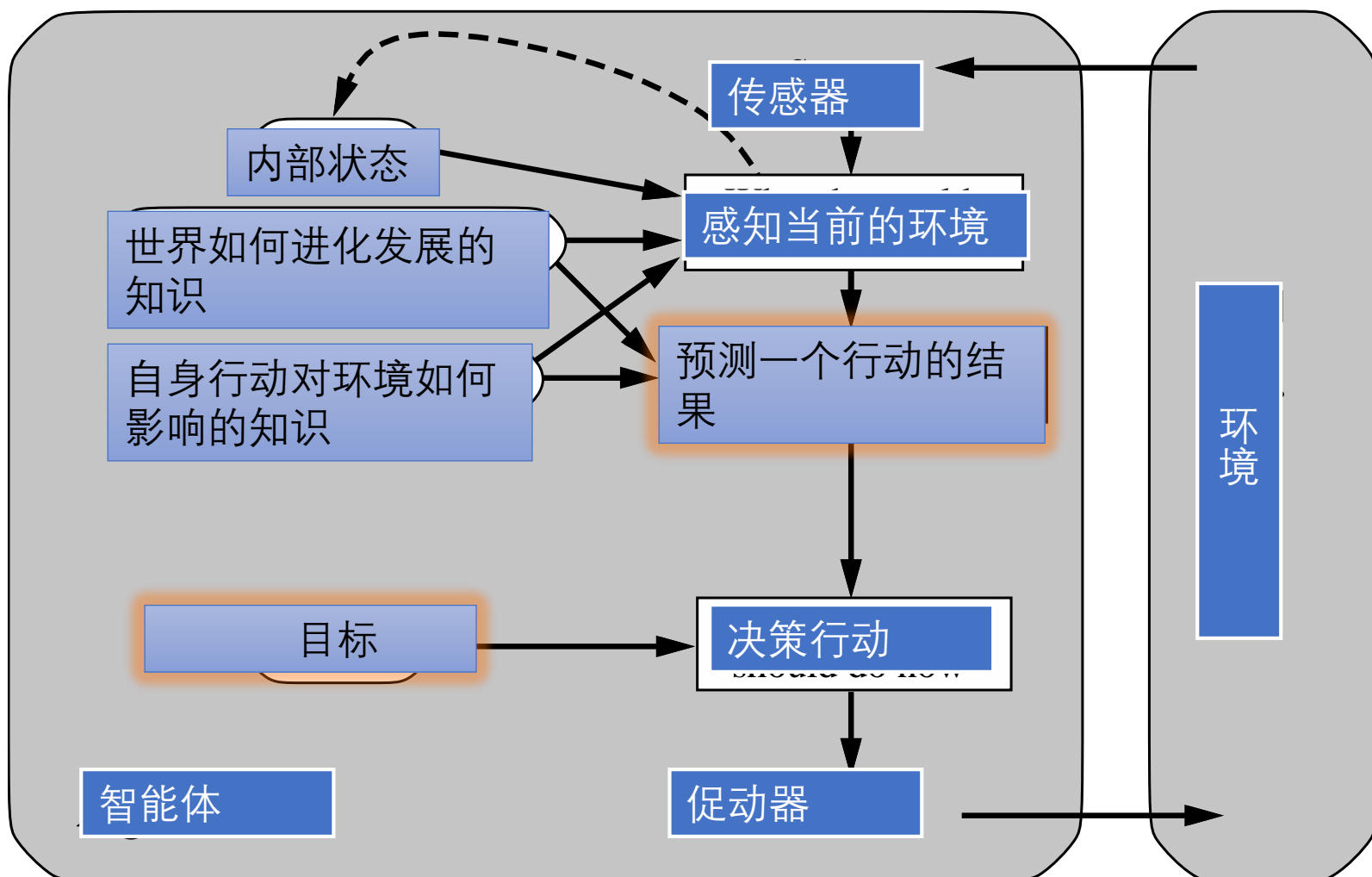
# 简单反射智能体



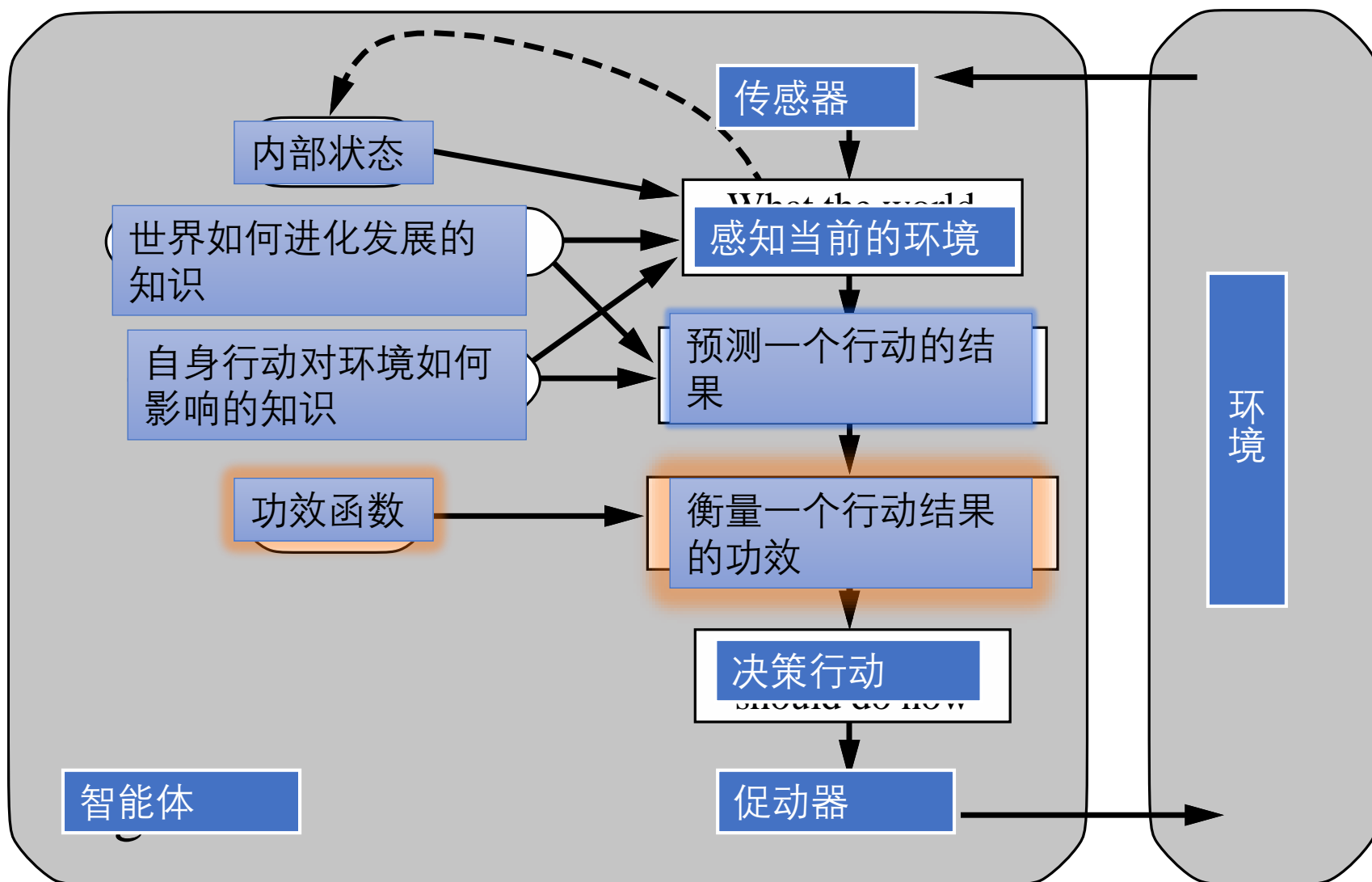
# 记录状态的反射智能体



# 基于目标的智能体



# 基于功效的智能体





# 总结

- 一个智能体通过自身的传感器和促动器，与环境作互动。
- 智能体函数（描述智能体在不同情况下采取的行动），由智能体程序在一台机器上来实现。
- PEAS 描述用来定义任务环境；准确的定义对智能体的设计很关键。
- 越困难的环境要求越复杂的智能体设计和复杂的结构。