祝恩

Agent

Agent及其理性

PEAS及环境特性

Agent程序的结构

Agent程序的部件如何运转

AGENT及其理性

Review

智能Agent开始出现：

- John Laird(1987)提出了**完整Agent**
  - 在现实环境中有连续传感器输入的Agent如何行动。例如，机器人、自动驾驶汽车。

Review

- 研究者们开始审视“完整Agent”。

1. 人们发现孤立的AI子领域需要综合起来。
2. 推理和规划系统必须能够处理不确定性。

- Agent观点导致AI与其他领域已经被拉得更靠近了。无人驾驶的进展来源于许多方法的混合，以及一定程度的高层次规划。

Agents {2.1}

- An agent is anything that can be viewed as
  - perceiving(感知) its environment through sensors(传感器)
  - acting(动作) upon that environment through actuators(执行器)
- Softbots, robots, humans, …

Agents {2.1}

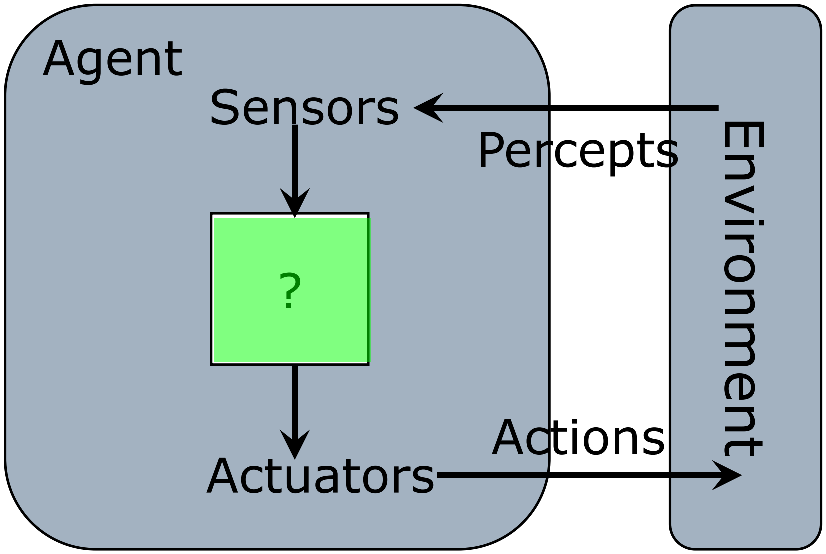
- Human agent
  - Sensors
    - eyes, ears, …
  - Actuators
    - hands, legs, mouth, …
- Robotic agent
  - Sensors
    - cameras and infrared range finders(红外测距仪)
  - Actuators
    - various motors

the structure of a general agent

Agents and Environments {2.1}

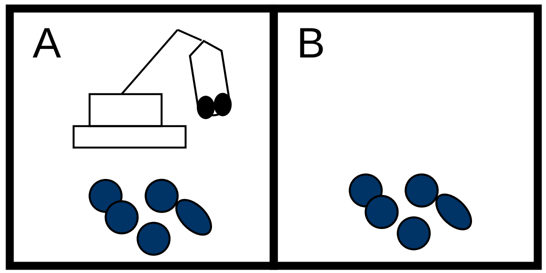
- An agent program is running on the (physical) architecture to produce the agent function.
- The agent function maps from percept histories to actions:

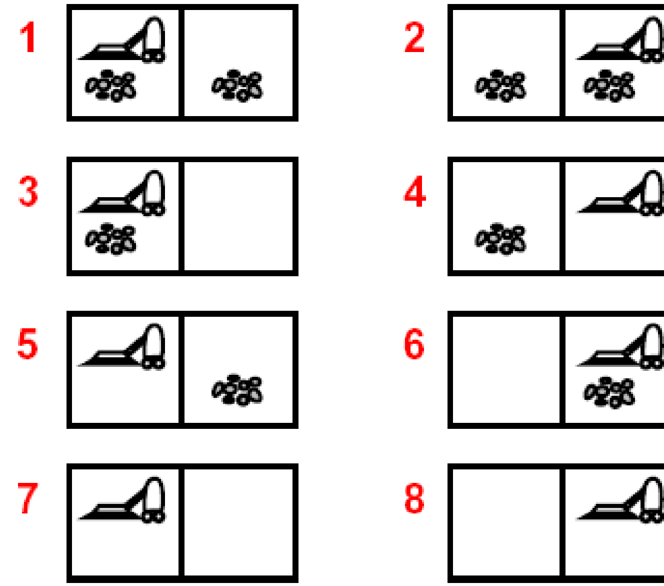
$$\mathcal{P}^* \rightarrow \mathcal{A}$$



- The agent's choice of action at any given instant can depend on the entire percept sequence observed to date.

Vacuum-cleaner agent {2.1}





- Percepts: location and contents, e.g. [A, dirty]
  - (Idealization: locations are discrete)
- Actions: LEFT, RIGHT, SUCK

A Reflex Vacuum-Cleaner {2.1}

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*

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。

rationality depends on…

Good behavior: The concept of Rationality {2.2}

- Rationality depends on
  - The performance measure that defines the criterion of success
  - The agent's prior knowledge of the environment
  - The actions that the agent can perform
  - The agent's percept sequence to date

Good behavior: The concept of Rationality {2.2.1}

- The definition of 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.*

Good behavior: The concept of Rationality {2.2.1}

- 假设：
  - 每个时间步清洁一个方块加1分，共1000步
  - 地形已知，灰尘分布和初始位置未知。吸尘清洁当前方格，干净的方格保持干净。Left和Right使Agent移动，但不出界。
  - 行动只有Left、Right、Suck
  - Agent能感知位置以及所在方格是否有灰尘
- 则Agent：有灰尘，则吸尘，否则移到另一格
- 是理性的吗？

Good behavior: The concept of Rationality {2.2.1}

- 假设：
  - 每个时间步清洁一个方块加1分，共1000步；每移动一步减1分
  - 地形已知，灰尘分布和初始位置未知。吸尘清洁当前方格，干净的方格保持干净。Left和Right使Agent移动，但不出界。
  - 行动只有Left、Right、Suck
  - Agent能感知位置以及所在方格是否有灰尘
- Agent：有灰尘，则吸尘，否则移到另一方格
- 是理性的吗？

Good behavior: The concept of Rationality {2.2.1}

- 这种情况下，一个更好的Agent应该在它确信所有的地方已经干净了以后不做任何事情。如果方格再次被弄脏了，该Agent应该不定期地检查并在必要的时候重新清洁。如果环境的地形未知的话，该Agent还需要去探查其它区域而不是固守方格A和B

Good behavior: The concept of Rationality {2.2.2}

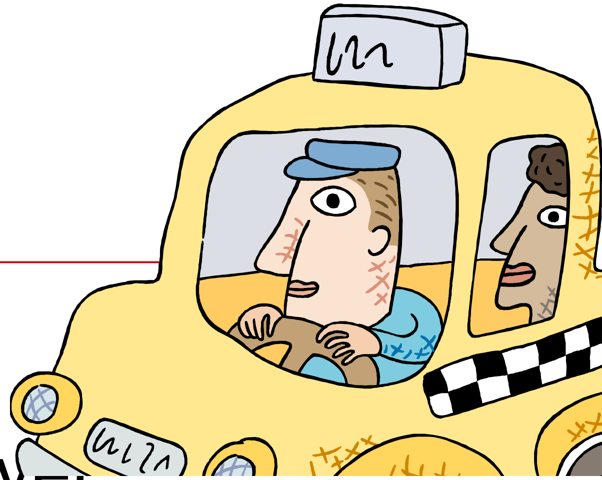
- 理性 ≠ 全知
  - 全知的Agent明确知道它的动作产生的实际结果
- 理性 ≠ 完美
  - 理性Agent使期望性能最大化；完美Agent使实际性能最大化。
- 理性Agent还应该
  - 收集信息、学习

PEAS及环境特性

The nature of environments {2.3.1}

- To design a rational agent we need to specify a task environment
- PEAS*: to* specify a task environment
  - *P*erformance measure
  - *E*nvironment
  - *A*ctuators
  - *S*ensors

PEAS: autobot {2.3.1}



- Specifying an automated taxi driver
  - Performance measure
    - safe, fast, legal, comfortable, maximize profits
  - Environment
    - roads, other traffic, pedestrians(行人), customers
  - Actuators
    - Steering(方向盘), accelerator(加速器), brake(刹车), signal, horn(喇叭)
  - Sensors
    - cameras, sonar(声纳), speedometer(速度计), GPS

PEAS

- Spam Filtering :
  - Performance measure: spam block
  - Environment: email client or server
  - Actuators: mark as spam, transfer messages
  - Sensors: emails (possibly across users), etc.

Properties of task environments {2.3.2}

- Fully vs. partially observable
  - can the sensors detect all aspects that are relevant to the choice of action?
- Deterministic vs. stochastic(随机)
  - if the next environment state is completely determined by the current state and the executed action?
- Episodic(片断式的) vs. sequential(序列式的)
  - if the agent’s experience can be divided into atomic steps and the choice of action depends only on the episode itself?

Properties of task environments {2.3.2}

- Static vs. dynamic
  - If the environment can change while the agent is choosing an action?
- Discrete vs. continuous
  - This distinction can be applied to the state of the environment, the way time is handled and to the percepts/actions of the agent.
- Single vs. multi-agent
  - Does the environment contain other agents who are also maximizing some performance measure that depends on the current agent's actions?

Agents and environments {2.3.2}

Task	Crossword puzzle	Poker	Taxi
Observable	fully	partially	Partially
Deterministic	deterministic	stochastic	stochastic
Episodic	sequential	sequential	sequential
Static	static	static	dynamic
Discrete	discrete	discrete	continuous
Agents	Single	multi	multi

The environment type largely determines the agent design

AGENT程序的结构

The structure of agents {2.4, 2.4.1}

- How does the inside of the agent work?
  - Agent = architecture + program
- All agents have the same skeleton
  - Input = current percepts,
  - Output = action
  - Program= manipulates input to produce output

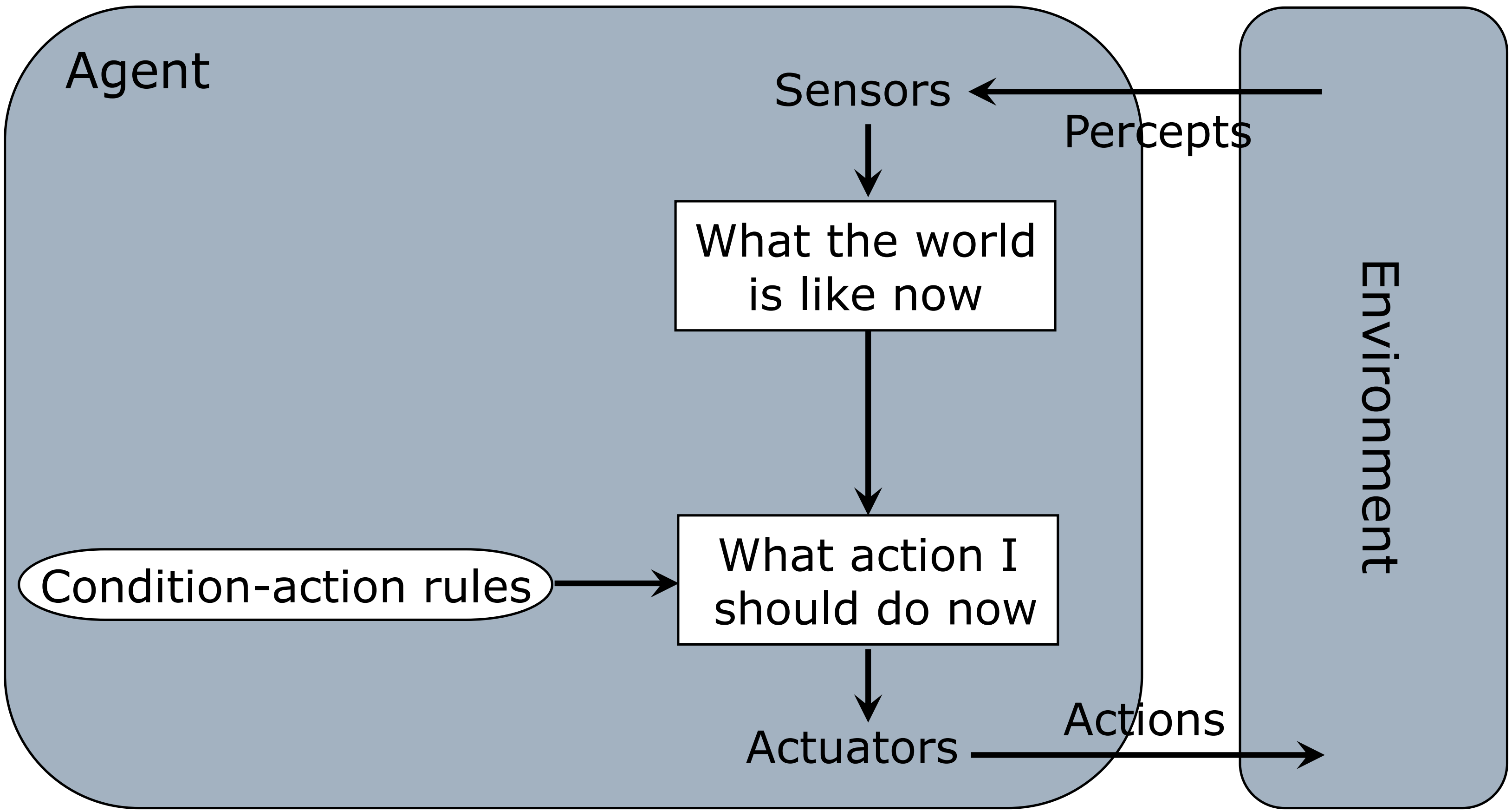
The structure of agents {2.4, 2.4.1}

- Four basic types in order of increasing generality
  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents
- All these can be turned into learning agents

Simple Reflex Agents {2.4.2}

- 最简单的Agent种类是**简单反射Agent**。这类Agent基于当前的感知选择行动，不关注感知历史。

Simple Reflex Agents {2.4.2}



基于当前的感知选择行动，不关注感知历史

Simple Reflex Agents {2.4.2}

```
function REFLEX-VACUUM-AGENTS([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
```

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

        state  <- INTERPRET-INPUT(percept);
        rule   <- RULE-MATCH(state, rules);
        action <- rule.ACTION;
        return action
```

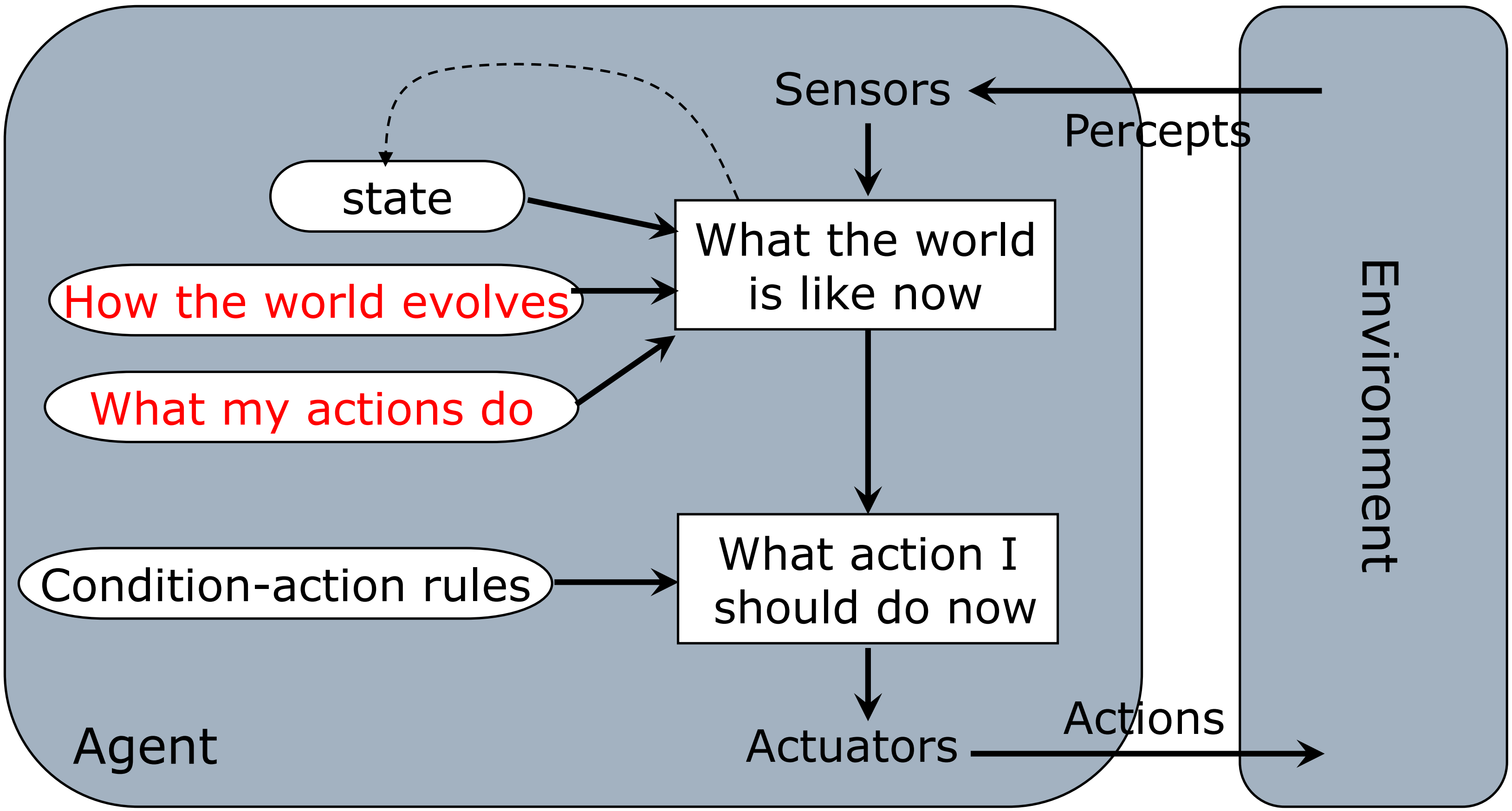
- 如何处理部分可观察的环境？
- 跟踪记录看不到的那部分。
- Agent需要知道世界是如何运转的，自身的行动是如何影响世界的。

Model-based reflex agents {2.4.3}

- 处理部分可观测环境：*跟踪记录现在看不到的那部分*。即，维持**内部状态。**
- 随时更新内部状态信息要求在Agent程序中加入两种类型的知识。首先，**世界是如何独立于Agent而发展的**。其次，**Agent自身的行动如何影响世界的**。这种关于“世界如何运转”的知识都被称为世界**模型**。

Model-based reflex agents {2.4.3}

- Maintain internal state to tackle partially observable environments.



Model-based reflex agents {2.4.3}

- Over time update state using world knowledge
  - How does the world change.
  - How do actions affect world. ⇒ *Model of World*

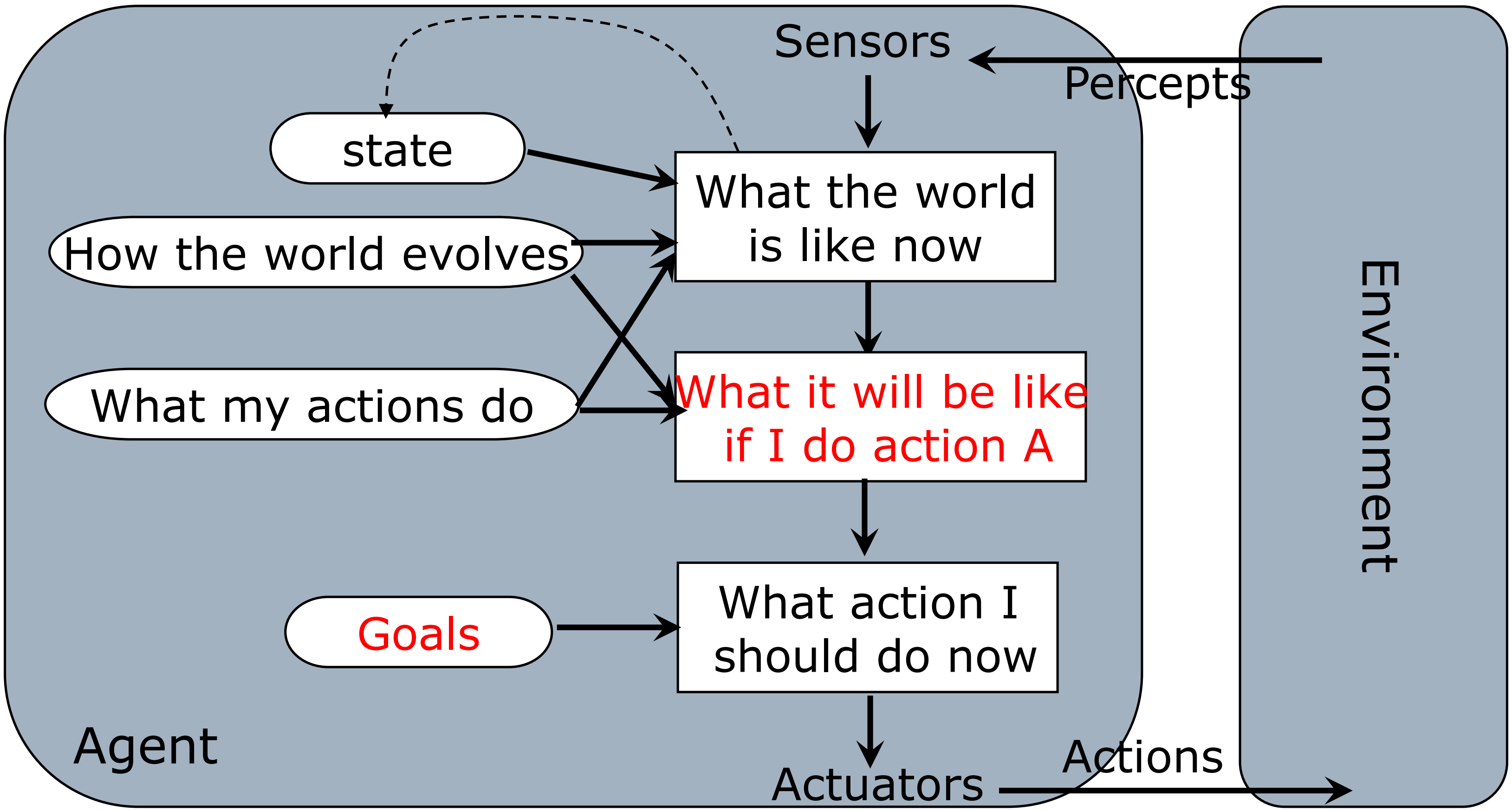
```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
    static: rules, a set of condition-action rules
            state, a description of the current world state
            action, the most recent action
            model, a description of how the next state depends on
                current state and action

    state ← UPDATE-STATE(state, action, percept, model)
    rule ← RULE-MATCH(state, rules)
    action ← rule.ACTION
    return action
```

- 如何达到自身的目标？
- 需要目标信息，以选择行动

Goal-Based Agents {2.4.4}

- usually first find plans then execute them

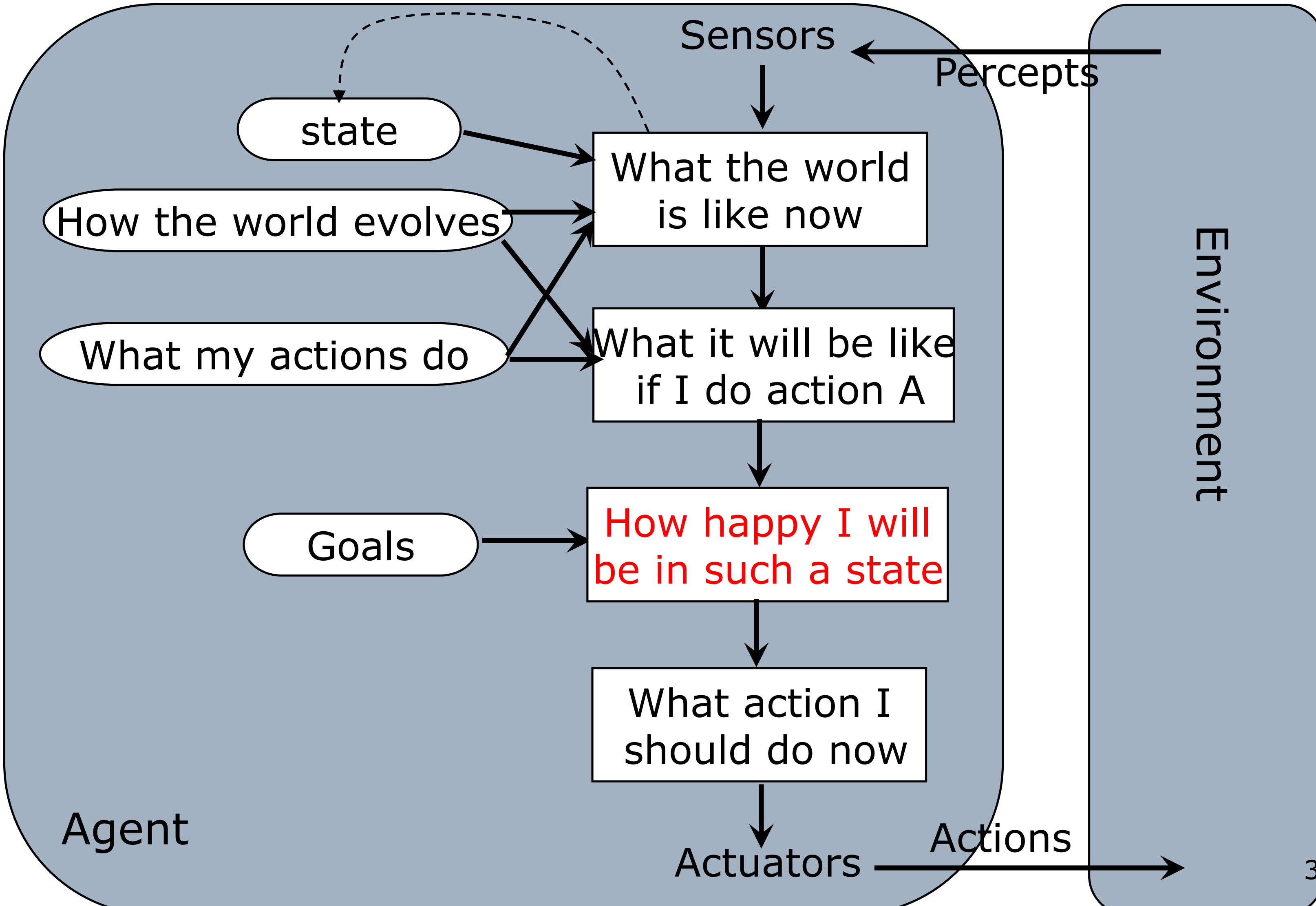


Goal-Based Agents {2.4.4}

- The agent needs a goal to know which situations are desirable
  - Things become difficult when long sequences of actions are required to find the goal.
- Typically investigated in search and planning research
- Major difference: future is taken into account
- Is more flexible since knowledge is represented explicitly and can be manipulated

- 如何生成高品质的行为？
- 需要知道哪些动作更快、更可靠、更安全

Utility-Based Agents {2.4.5}



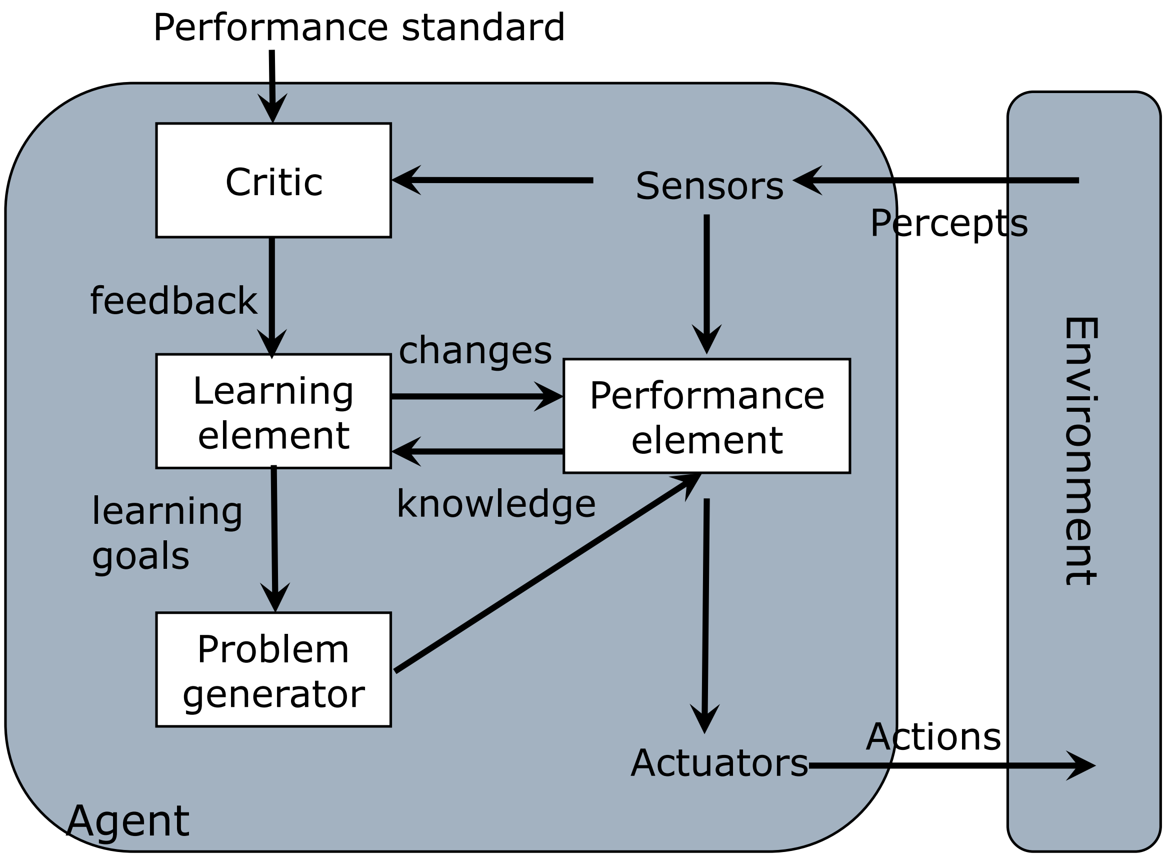
Utility-Based Agents {2.4.5}

- Certain goals can be reached in different ways.
  - Some are better, have a higher utility.
- Utility function maps a (sequence of) state(s) onto a real number.
- Improves on goals:
  - Selecting between conflicting goals
  - Select appropriately between several goals based on likelihood of success.
- Remember: rationality depends on
  - Performance measure
  - Agent's (prior) knowledge
  - Agent's percepts to date
  - Available actions

- Agent程序如何形成，如何获得选择动作的能力？

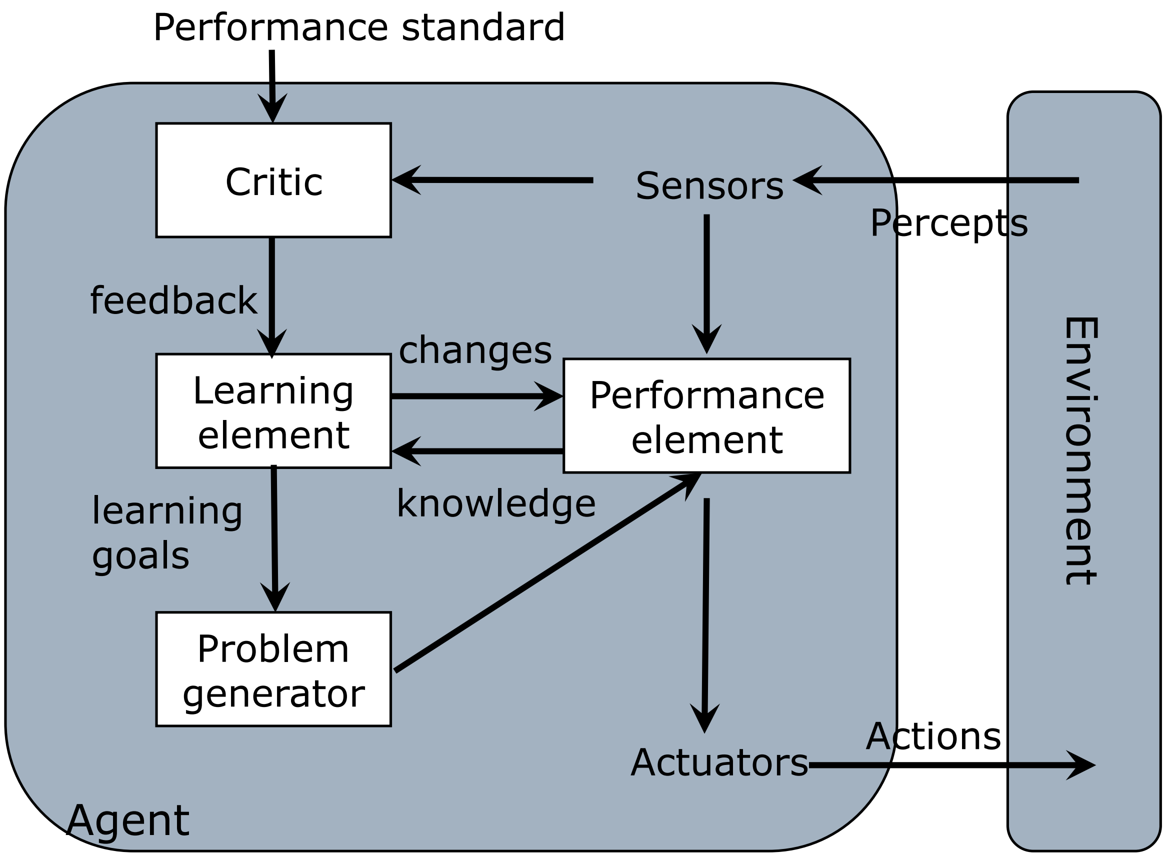
Learning agent {2.4.6}

- 4 componets:
  - *Performance element* *执行部件*
  - *Learning element.* 学习部件
  - *Critic.* 评判部件
  - *Problem generator* *问题生成器*



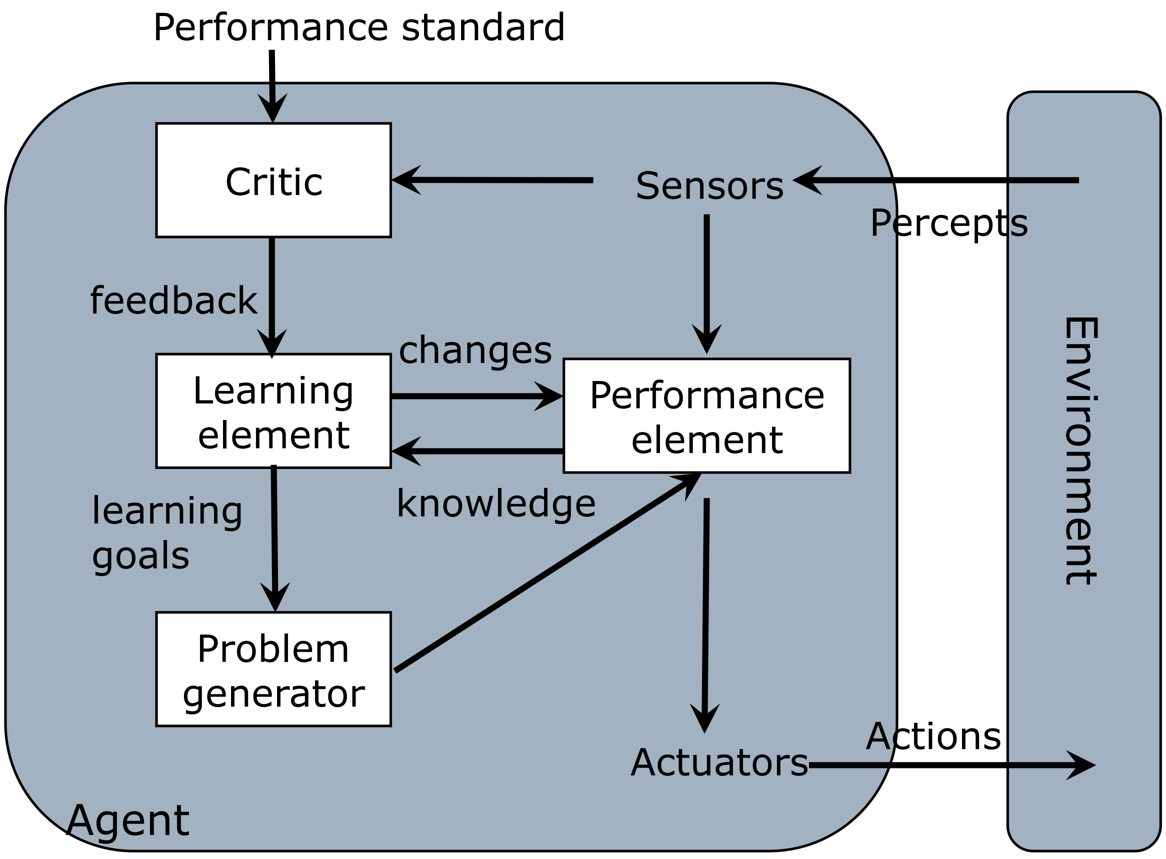
Learning agent {2.4.6}

- *Performance element*: selecting actions based on percepts.
- 执行部件就是前面考虑的整个Agent，接受感知，选择动作。



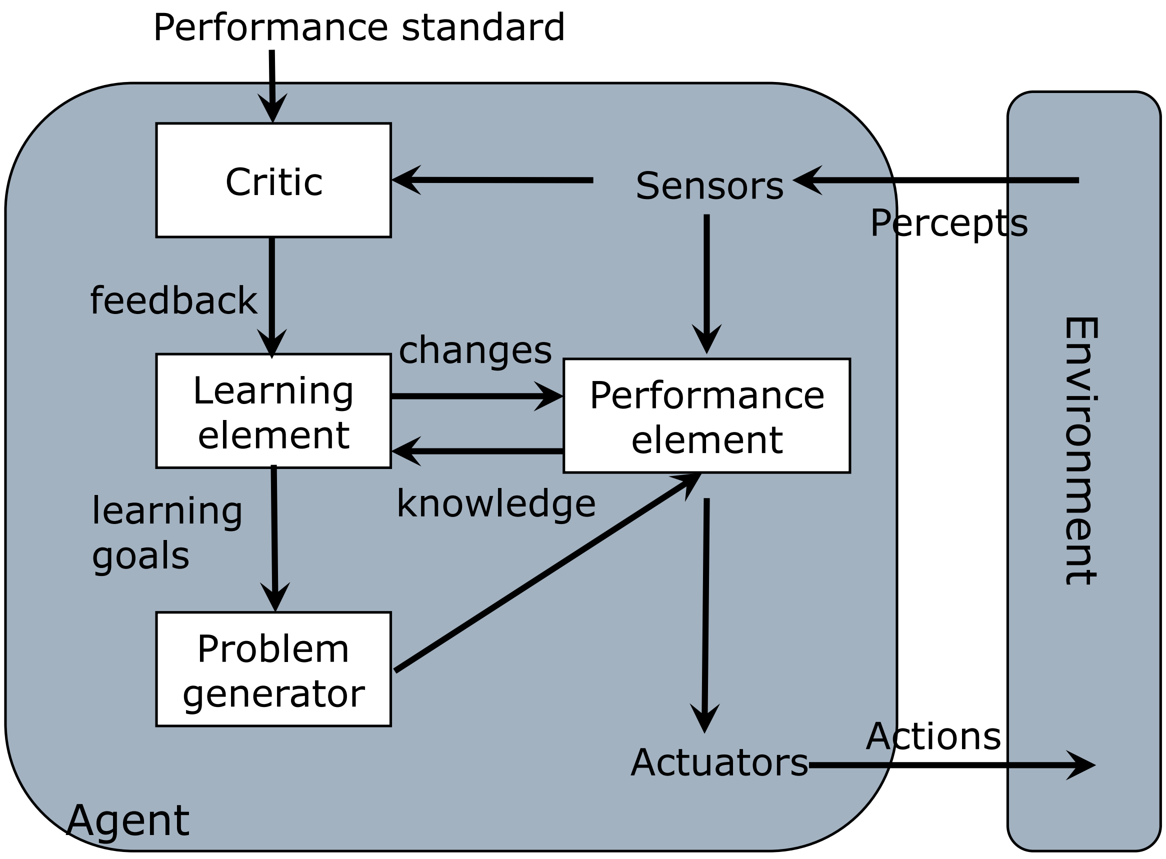
Learning agent {2.4.6}

- *Learning element*: introduce improvements in performance element.
- 学习部件根据评判部件的反馈评价Agent做得如何，从而确定如何修改执行部件。



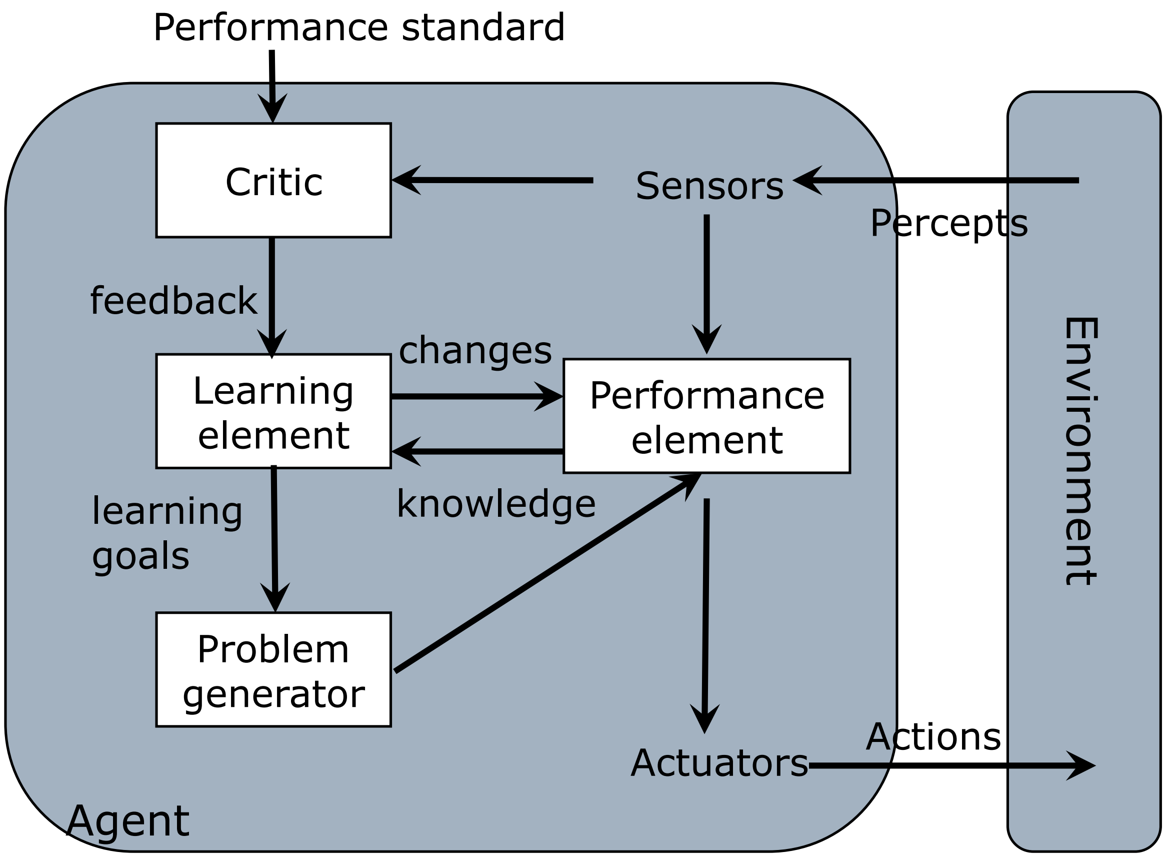
Learning agent {2.4.6}

- Critic
- 评判部件根据性能标准告知学习部件Agent的运行情况。当“将军”发生了，评判部件告知学习部件：好事情发生了
- 性能标准是固定的，Agent不应该修改性能标准来适应自己的行为



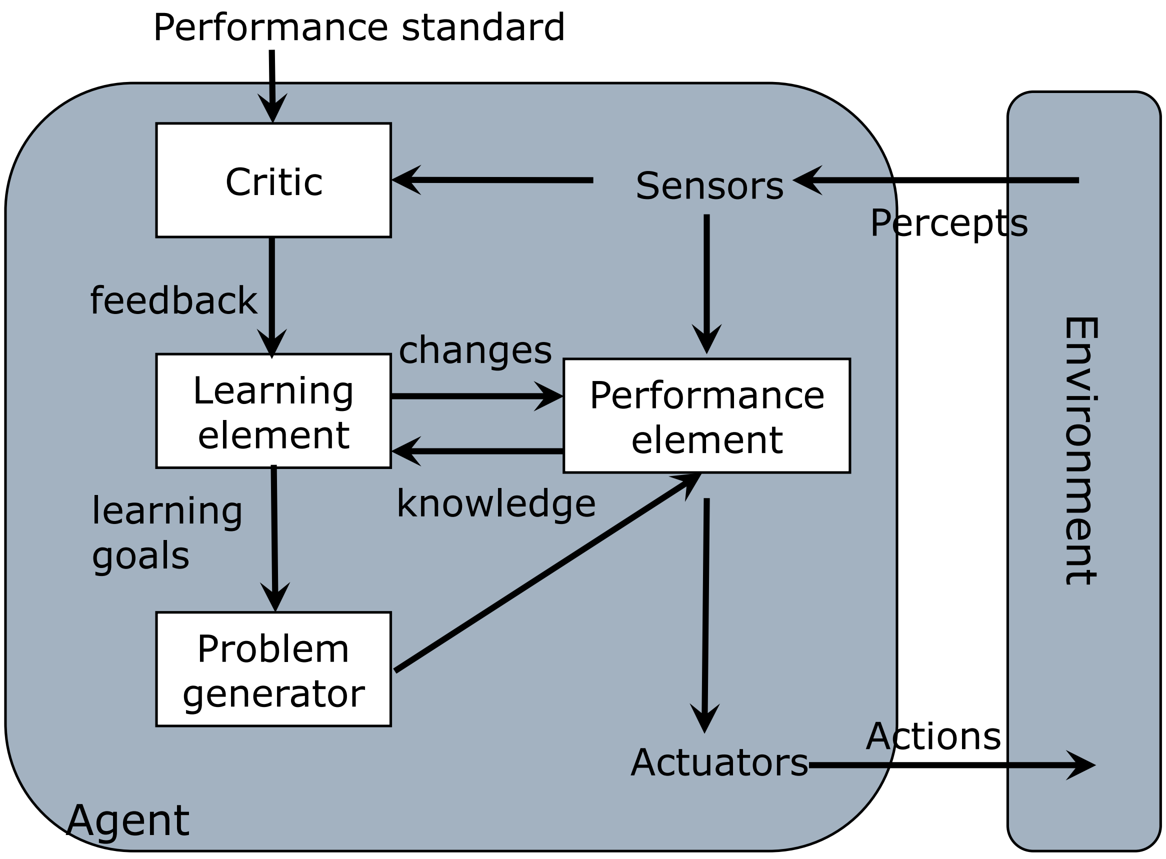
Learning agent {2.4.6}

- *Problem generator*: suggests actions that will lead to new and informative experiences.
- 如果执行部件自行其是，它会一直根据已知的知识采取最佳行动
- 问题生成器可向执行部件建议探索性行动，短期内可能是次优的行动，但长远而言可能是更好的行动



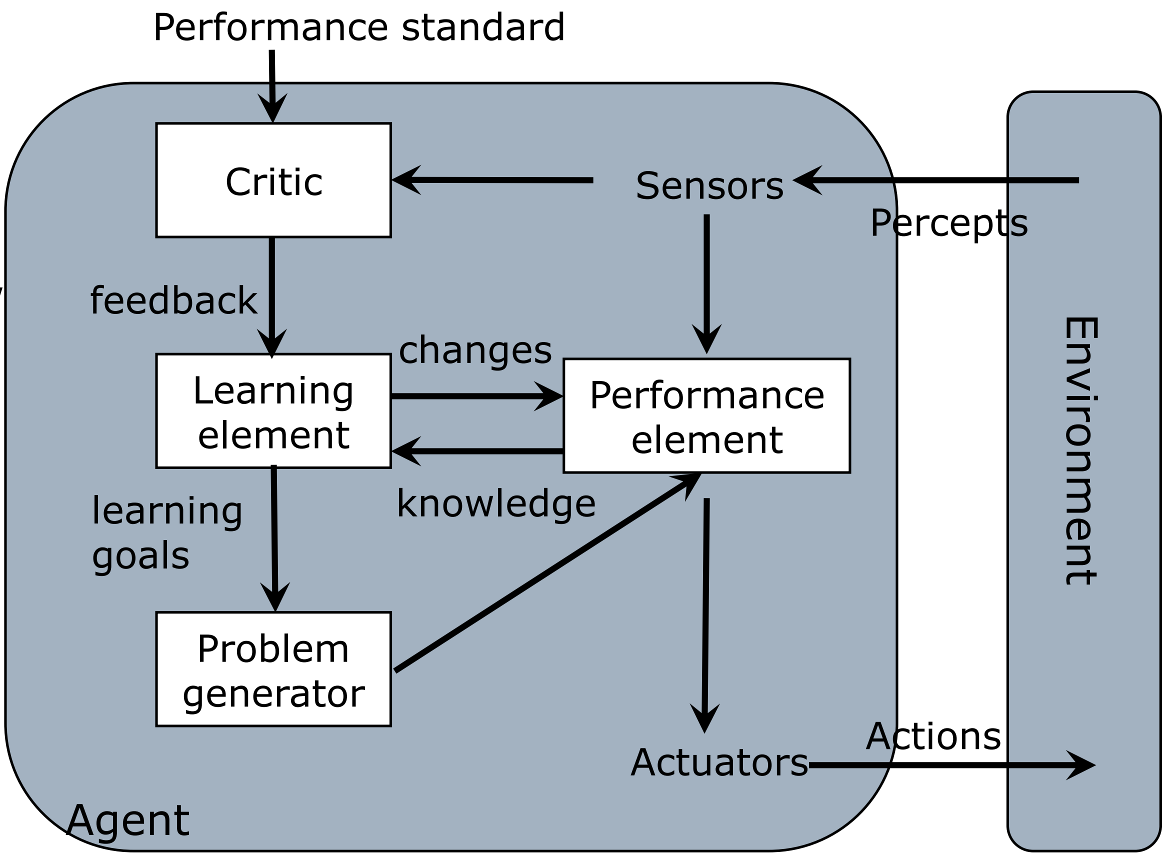
Learning agent {2.4.6}

- 学习部件的设计很大程度上依赖于执行部件的设计。当设计学习特定技能的Agent时，第一个问题不是“我怎样才能让Agent学到这个技能？”，而是“Agent一旦学会了这个技能，需要何种执行部件来行使该能力？”对于给定的Agent设计，可以构造学习机制来改进Agent的各个部分。



Learning agent {2.4.6}

- 自动驾驶
- 执行部件：使用执行部件在公路上行驶。执行部件包含选择驾驶行动的全部知识和过程集合。
- 评判部件：观察评价世界，告知学习部件。不打转向灯变道，听到后车喇叭声。
- 学习部件：制定修改规则。如果是不好的行动，如何修改执行部件。
- 问题生成器：例如提议在不同路面试验一下刹车效果。



- Agent的各个部件是如何工作的？

How the components of agent program work? {2.4.7}

- Agent程序包含回答以下问题的部件：
- “世界现在怎样了？” “我现在应该采取什么行动了？” “我的行动结果怎么样”

How the components of agent program work? {2.4.7}

- 处理“我的行动后果如何”的组件描述了发生在环境中的变化，将之视为行动的后果。
- 如何表示状态及其转换？

- 沿着复杂度和表达能力增长的轴线有三种表示：原子表示、要素化表示、结构化表示

How the components of agent program work? {2.4.7}

- **原子表示**：罗马尼亚问题中，状态是城市的名字——原子——没有内部结构。
- 使用原子表示：搜索和博弈（ch03-05)隐马尔科夫模型(ch15)马尔科夫决策过程(ch17)

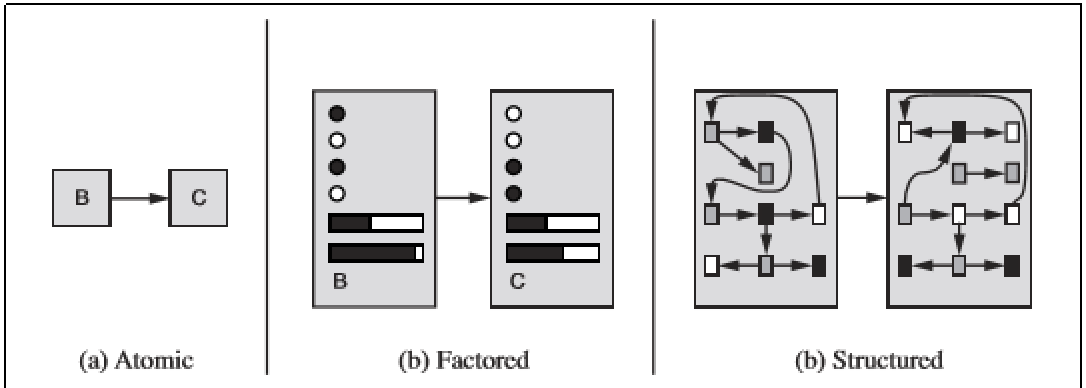


Figure 2.16 Three ways to represent states and the transitions between them. (a) Atomic representation: a state (such as B or C) is a black box with no internal structure; (b) Factored representation: a state consists of a vector of attribute values; values can be Boolean, real-valued, or one of a fixed set of symbols. (c) Structured representation: a state includes objects, each of which may have attributes of its own as well as relationships to other objects.

How the components of agent program work? {2.4.7}

- **要素化表示**：状态表示为变量或特征的集合
- 使用要素化表示：约束满足(ch06)命题逻辑(ch07)规划(ch10ch11)贝叶斯网(ch13-16)机器学习(ch18-21)

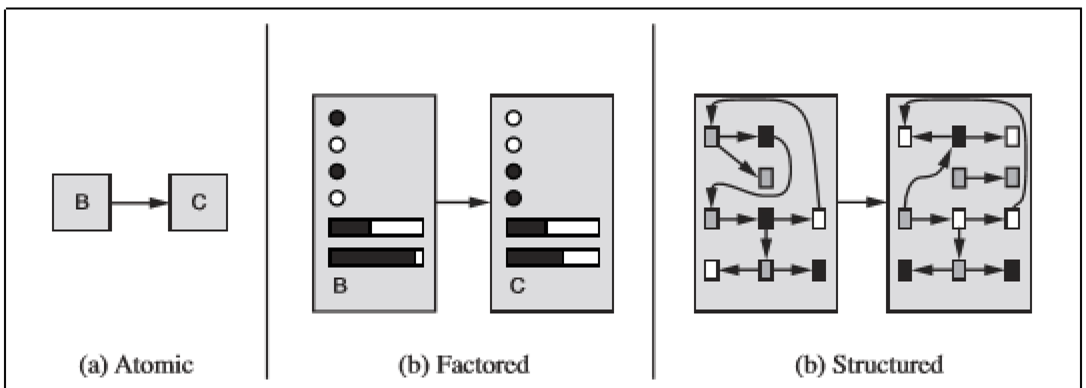


Figure 2.16 Three ways to represent states and the transitions between them. (a) Atomic representation: a state (such as B or C) is a black box with no internal structure; (b) Factored representation: a state consists of a vector of attribute values; values can be Boolean, real-valued, or one of a fixed set of symbols. (c) Structured representation: a state includes objects, each of which may have attributes of its own as well as relationships to other objects.

How the components of agent program work? {2.4.7}

- **结构化表示**：世界中有事物，事物间有关联
- 使用结构化表示：关系数据库、一阶逻辑(ch8/9/12)一阶概率模型(ch14)基于知识的学习(ch19)自然语言理解(ch22/23)

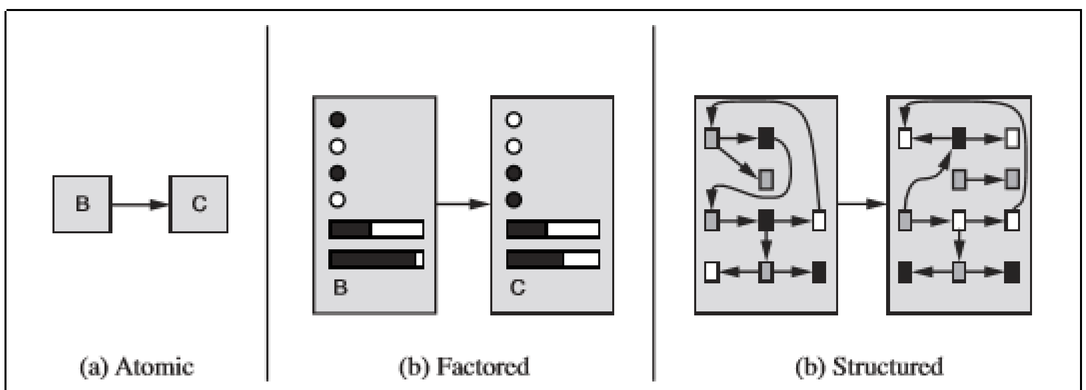


Figure 2.16 Three ways to represent states and the transitions between them. (a) Atomic representation: a state (such as B or C) is a black box with no internal structure; (b) Factored representation: a state consists of a vector of attribute values; values can be Boolean, real-valued, or one of a fixed set of symbols. (c) Structured representation: a state includes objects, each of which may have attributes of its own as well as relationships to other objects.

How the components of agent program work? {2.4.7}

- 表达能力越强，表示就越简洁。
- 表达能力越强，推理和学习就越复杂。

- 为了找到更具表达力的表示，避免它们的缺点，现实世界的智能系统可能需要同时在轴线上的所有点上操作

Summary {2.5}

- Agents interact with environments through actuators and sensors.
  - The agent function describes what the agent does in all circumstances
  - The agent program calculates the agent function
- A perfectly rational agent maximizes expected performance
- PEAS descriptions define task environments, environments are categorized along several dimensions.
- Architectures of Agent program

Thanks!