Defining Intelligent Agents

- Intelligent Agents (IA)
- Environment types
- IA Behavior
- IA Structure
- IA Types

What is an (Intelligent) Agent?

- An over-used, over-loaded, and misused term.

- Anything that can be *viewed as* **perceiving** its **environment** through **sensors** and **acting** upon that environment through its **effectors** to maximize progress towards its **goals**.

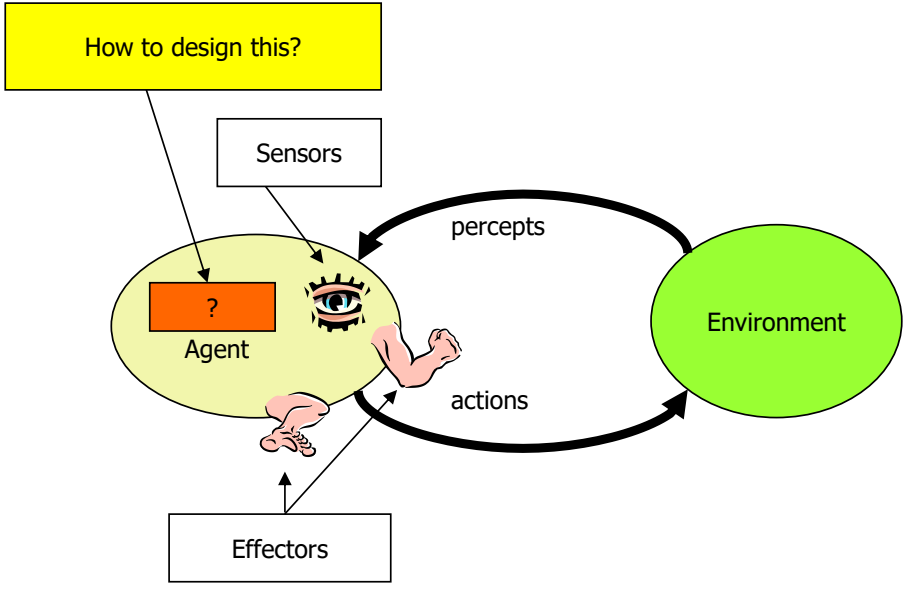
What is an (Intelligent) Agent?

- **PAGE** (Percepts, Actions, Goals, Environment)

- Task-specific & specialized: well-defined goals and environment

- The notion of an agent is meant to be a tool for analyzing systems, It is not a different hardware or new programming languages

Agent and Environment Interactions



A Windshield Wiper Agent (Cont'd)

- Goals: Keep windshields clean & maintain visibility
- Percepts: Raining, Dirty
- Sensors: Camera (moist sensor)
- Effectors: Wipers (left, right, back)
- Actions: Off, Slow, Medium, Fast
- Environment: Inner city, freeways, highways, weather …

Interactions Among Agents

Collision Avoidance Agent (CAA)

- Goals: Avoid running into obstacles
- Percepts: Obstacle distance, velocity, trajectory
- Sensors: Vision, proximity sensing
- Effectors: Steering Wheel, Accelerator, Brakes, Horn, Headlights
- Actions: Steer, speed up, brake, blow horn, signal (headlights)
- Environment: Freeway

Lane Keeping Agent (LKA)

- Goals: Stay in current lane
- Percepts: Lane center, lane boundaries
- Sensors: Vision
- Effectors: Steering Wheel, Accelerator, Brakes
- Actions: Steer, speed up, brake
- Environment: Freeway

Conflict Resolution by Action Selection Agents

- **Override:** CAA overrides LKA
- **Arbitrate:** if Obstacle is Close then CAA else LKA
- **Compromise:** Choose action that satisfies both agents
- Any combination of the above
- **Challenges:** Doing the right thing at the right time

The Right Thing = The Rational Action

- **Rational Action:** The action that maximizes the expected value of the performance measure given the percept sequence to date

  - Rational = Best — Yes, to the best of its knowledge
  - Rational = Optimal — Yes, to the best of its abilities (incl. its constraints)
  - Rational ≠ Omniscience
  - Rational ≠ Clairvoyant (supernatural)
  - Rational ≠ Successful

三、

Behavior and Performance of IAs

感知之间的所有时刻的完整历史 (input variables)

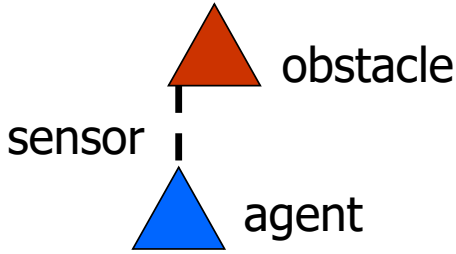
- **Perception** (sequence) **to Action Mapping:** $f: \mathcal{P}^* \rightarrow \mathcal{A}$
  - **Ideal mapping:** specifies which actions an agent ought to take at any point in time
  - **Implementation:** Look-Up-Table, Closed Form, Algorithm, etc.

- **Performance measure:** a *subjective* measure to characterize how successful an agent is (e.g., speed, power usage, accuracy, money, etc.)

- (degree of) **Autonomy:** to what extent is the agent able to make decisions and take actions on its own?

"Look-Up-Table" Agent

Distance	Action
10	No action
5	Turn left 30 degrees
2	Stop



Closed Form

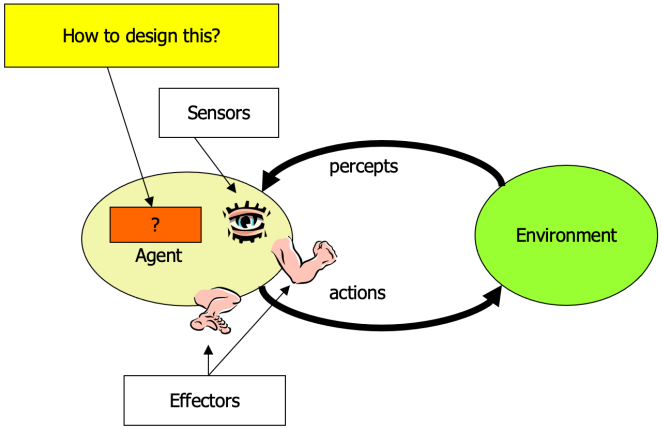
- Output (degree of rotation) = F(distance)
- E.g., F(d) = 10/d (distance cannot be less than 1/10)

How is an Agent different from other software?

Is A/C Thermostats an Agent? √
Is Zoom an agent? X

- Agents are **autonomous**, that is, they act on behalf of the user
- Agents contain some level of **intelligence**, from fixed rules to learning engines that allow them to adapt to changes in the environment
- Agents don't only act **reactively**, but sometimes also **proactively**
- Agents have **social ability**, that is, they communicate with the user, the system, and other agents as required
- Agents may also **cooperate** with other agents to carry out more complex tasks than they themselves can handle
- Agents may **migrate** from one system to another to access remote resources or even to meet other agents

Types of Environments



- Characteristics
  - Accessible vs. Inaccessible
  - Deterministic vs. Nondeterministic
  - Episodic vs. Non-episodic
  - Hostile vs. Friendly
  - Static vs. Dynamic
  - Discrete vs. Continuous

Environment Types

- Characteristics

(1) Accessible (observable) vs. inaccessible (partial observable)
  - Accessible: sensors give access to **complete** state of the environment.

same action + same state ⇒ same result

(2) Deterministic vs. nondeterministic
  - Deterministic: the next state can be determined based on the current state and the action.

don't need remember history need

(3) Episodic vs. non-episodic (Sequential)
  - Episode: each perceive and action pairs
  - In episodic environments, the quality of action does not depend on the previous episode and does not affect the next episode.
  - Example: Episodic: mail sorting system; non-episodic: chess

Environment Types

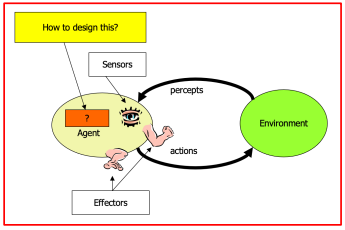
- Characteristics
  - Hostile vs. friendly
  - Static vs. dynamic
    - Dynamic if the environment changes during deliberation
  - Discrete vs. continuous
    - Chess vs. driving

2. Environment types

Environment	Accessible	Deterministic	Episodic	Static	Discrete
Operating System	Yes	Yes	No	No	Yes
Virtual Reality	Yes	Yes	Yes/no	No	Yes/no
Office Environment	No	No	No	No	No
Mars	No	Semi	No	Semi	No

The environment types largely determine the agent design.

Structure of Intelligent Agents



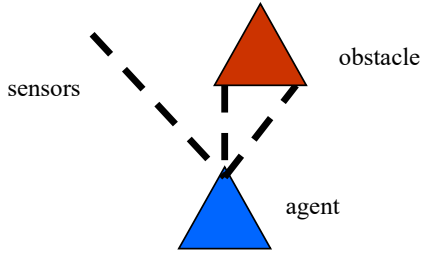
- **Agent** = architecture + program
- **Agent Program:** the implementation of $f: \mathcal{P}^* \rightarrow \mathcal{A}$, the agent's perception-action mapping

```
function Skeleton-Agent(Percept) returns Action
        memory ← UpdateMemory(memory, Percept)
        Action ← ChooseBestAction(memory)
        memory ← UpdateMemory(memory, Action)
return Action
```

- **Architecture:** a device that can execute the agent program (e.g., general-purpose computer, specialized device, beobot, etc.)

Using a look-up-table to encode $f : \mathcal{P}^* \rightarrow \mathcal{A}$

- Example: Collision Avoidance
  - Sensors: 3 proximity sensors
  - Effectors: Steering Wheel, Brakes



- How to generate: for each $p \in \mathcal{P}_l \times \mathcal{P}_m \times \mathcal{P}_r$ generate an appropriate action, $a \in S \times \mathcal{B}$

- How large: size of table = # possible percepts times # possible actions = $|\mathcal{P}_l|$ $|\mathcal{P}_m|$ $|\mathcal{P}_r|$ $|S|$ $|\mathcal{B}|$
  E.g., P = {close, medium, far}^{3}
  A = {left, straight, right} $\times$ {on, off}
  then size of table = 27 rows
- Total possible combinations (ways to fill table): 27*3*2=162

- How to select action? Search.

Agent Types

- Reflex agents
- Reflex agents with internal states
- Goal-based agents
- Utility-based agents
- Learning agents

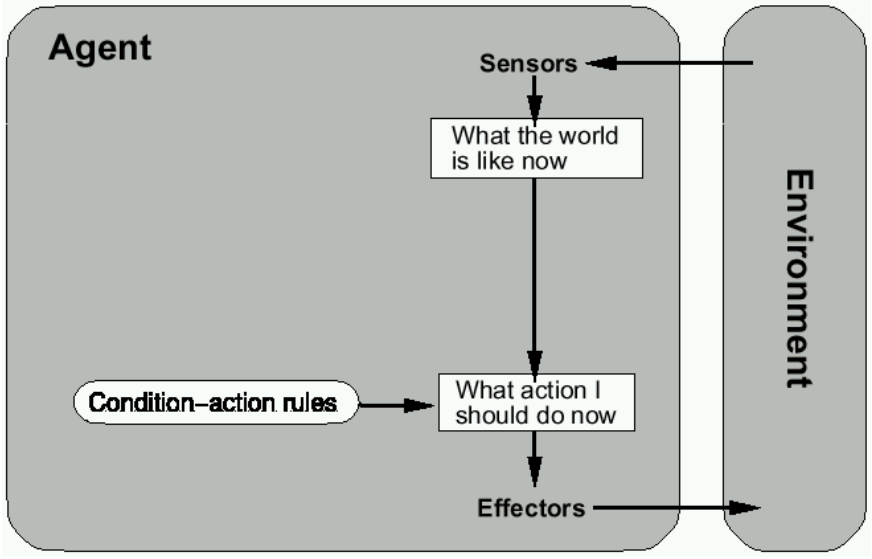
Agent Types

- Reflex agents
  - Reactive: No memory
- Reflex agents with internal states
  - W/o previous state, may not be able to make decision
    - E.g. brake lights at night.
- Goal-based agents
  - Goal information needed to make decision

Agent Types

- Utility-based agents
  - How well can the goal be achieved (degree of happiness)
  - What to do if there are conflicting goals?
    - Speed and safety
  - Which goal should be selected if several can be achieved?
- Learning agents
  - How can I adapt to the environment?
  - How can I learn from my mistakes?

Reflex Agents



- simple reflex agent: 简单反射agent基于当前感知选择操作。如下图所示，类似于人类或动物的条件反射，这种agent会有一系列相应的简单规则进行操作。如下面的示意图和程序伪代码所示。布尔电路的逻辑门就可以轻易将此实现。缺点是，此类agent结构意味着它仅仅对当前的感知有反应，也就是说，只有在环境完全可观测时可用。

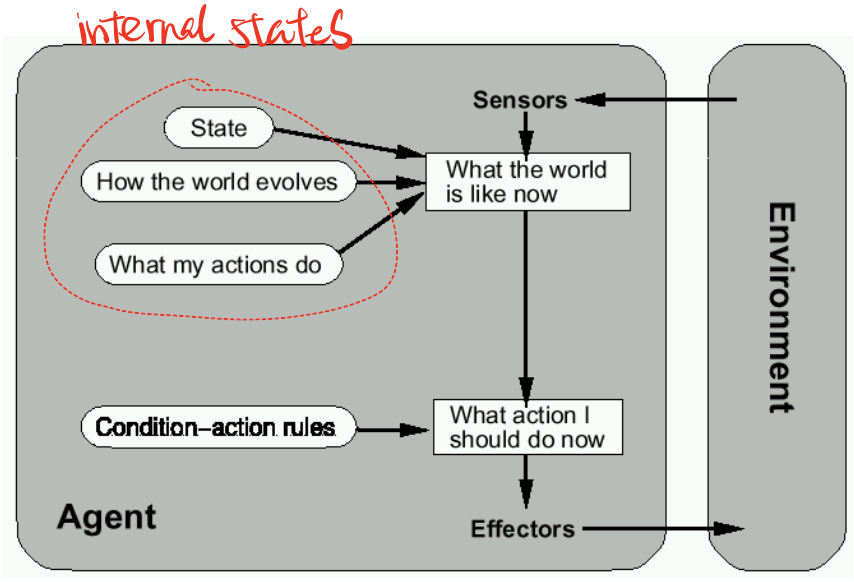
Reactive Agents

- Reactive agents do not have internal symbolic models.
- Act by stimulus-response to the current state of the environment.
- Each reactive agent is simple and interacts with others in a basic way.
- Complex patterns of behavior emerge from their interaction.

- **Benefits:** robustness, fast response time
- **Challenges:** scalability, how intelligent? how do you debug them?

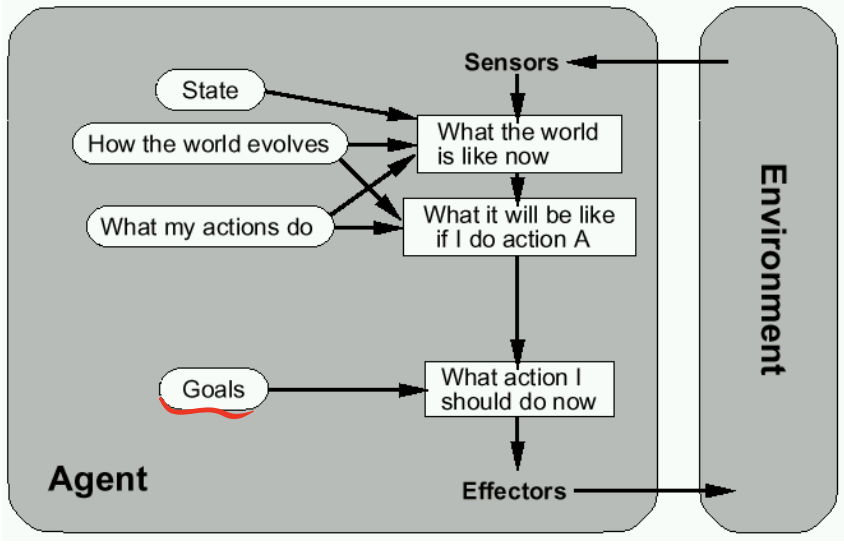
2、Reflex Agents with Internal States

- Model-based agent: 处理部分可观测性一个有效的方法是让智能体跟踪它当前看不到的世界,也就是维护依赖于感知历史的内部状态。比如自动驾驶里面记录前几帧的视频数据可以帮助判断一些问题。在这种agent中，我们需要了解在感知到的信息里面，世界如何随时间变化：agent自身行为对世界有什么影响？世界独立于agent的影响？
  总之该模型会将当前感知信息和过去内部状态结合来进行判断。这种模型的缺点是：部分可观测性的环境下agent仍无法准确确定当前状态。



3、Goal-Based Agents

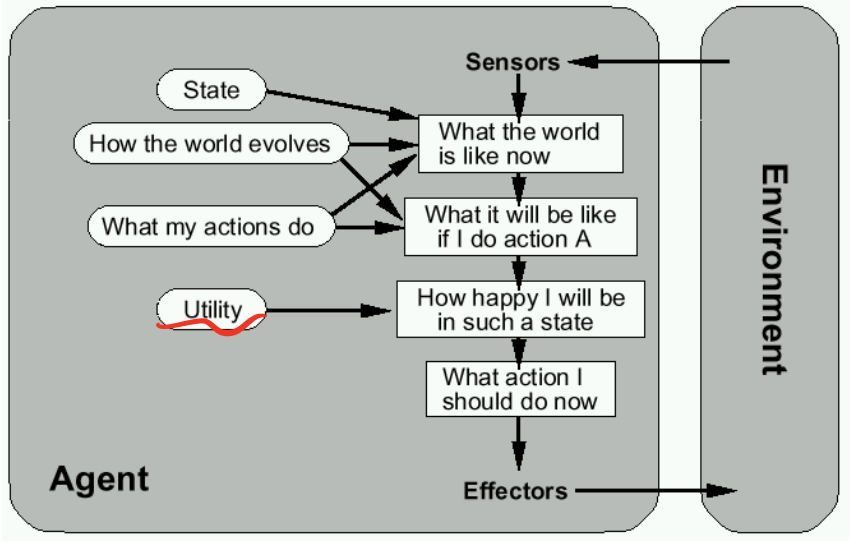
- Goal-based agent:除了以上两种模型的判断外，这类 agent意味着我们还需要了解我们的目标信息是什么。它要记录到目标地点的信息序列，并采取行动。 它的优点是灵活和可学习，但它效率比较低，如果多个目标冲突时就你难以为继了。



4. Utility-Based Agents

- Utility-based agent: 基于效用的智能体不只是在好与不好之间做选择，而是采取对智能体利益的程度（Utility）来进行判断，而现实世界往往是部分可观察性和不确定性的，因此基于效用的agent会使用效用函数来最大化预期效用。效用函数也可以被看做前面性能度量的某种内部实现。

  由于计算复杂性的原因，很多时候还是无法完全实现智能体的“理性”。



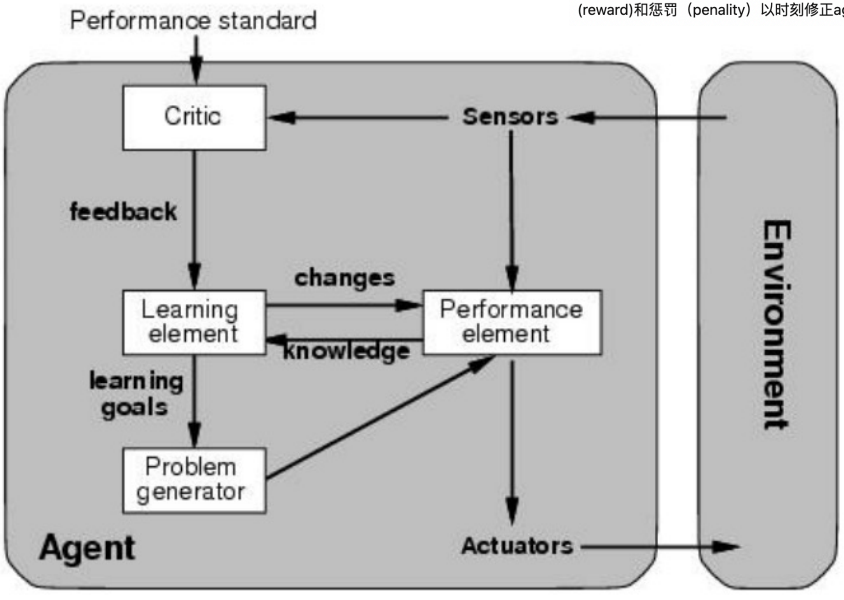
Learning Agents

目前，以上四种agent都可以被构建为学习型agent（learning agent)学习的优势在于，它允许agent在最初未知的环境中操作，并使其变得比开始的时候更强大。 一个学习agent可以四个概念元件：学习元件(Learing element)，性能元件(Performance element)，Critic， 问题生成器(Problem generator) 。"学习元件负责改进提高。性能元件负责选择外部行动，接受感知信息并决策。评判元件根据固定的性能指标告诉学习元件Agent的运转情况。问题产生器负责对新的和有信息的经验的行动提议，建议探索性行动（次优的行动）"[1]。

另外，在以上的学习agent里，性能标准（Performance standard）会将传入感知区分为奖励(reward)和惩罚（penality）以时刻修正agent。

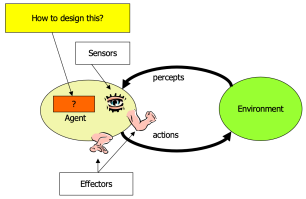
Critic: Determines outcomes of actions and gives feedback

Learning element: Takes feedback from critic and improves performance element



Problem generator: creates new experiences to promote learning

Summary on Intelligent Agents



- **Intelligent Agents:**
  - Anything that can be *viewed as* **perceiving** its **environment** through **sensors** and **acting** upon that environment through its **effectors** to maximize progress towards its **goals**.
  - PAGE (Percepts, Actions, Goals, Environment)
  - Described as a Perception (sequence) to Action Mapping: $f: \mathcal{P}^* \rightarrow \mathcal{A}$
  - Using look-up-table, closed form, etc.

- **Agent Types:** Reflex, state-based, goal-based, utility-based, learning

- **Rational Action:** The action that maximizes the expected value of the performance measure <u>given the percept sequence to date</u>