Artificial Intelligence *CS361*

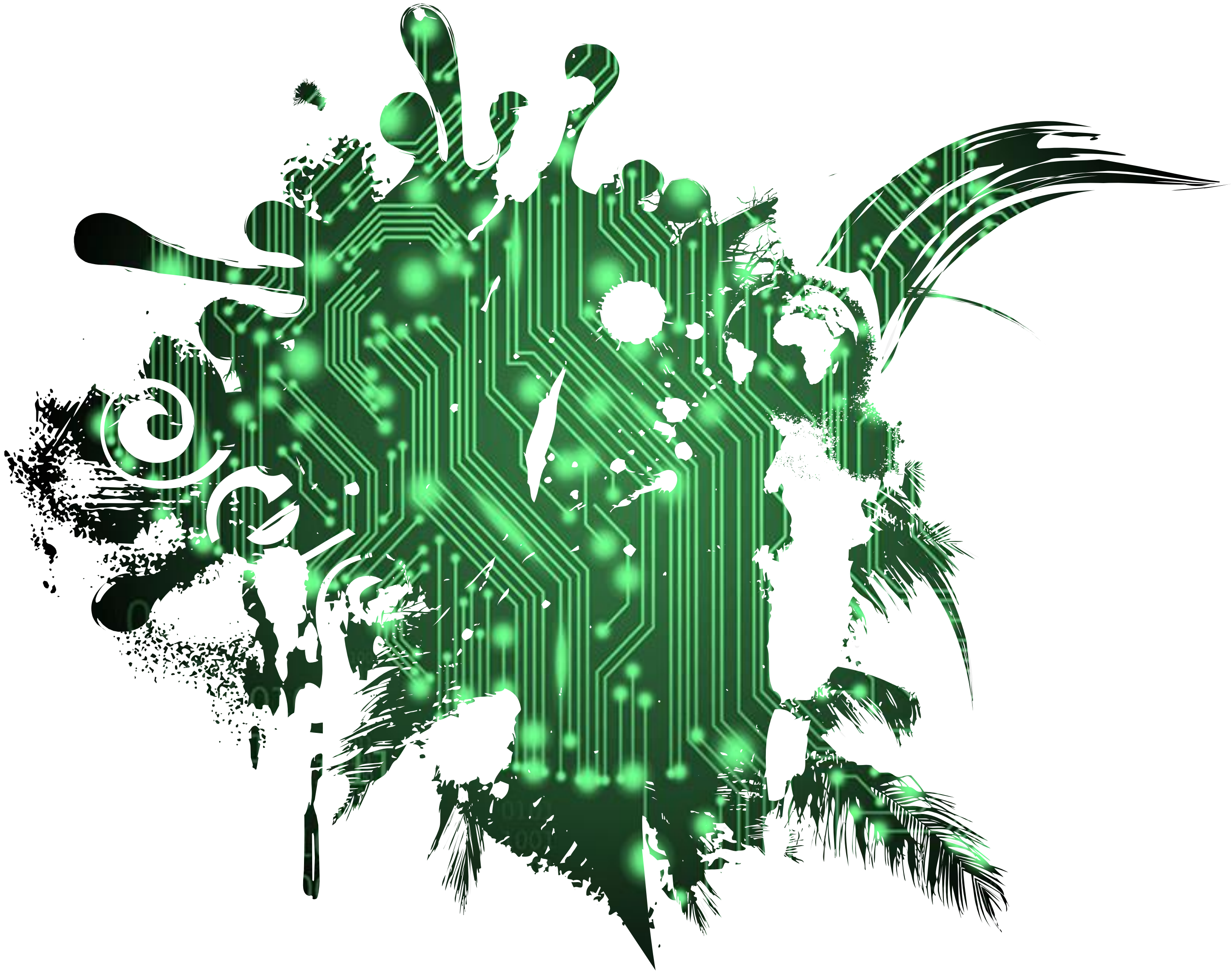
Intelligent Agents

Chapter 2

Contents

» Intelligent Agents (IA)
» Environment types
» IA Structure
» IA Types

Intelligent Agents (IA)



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.

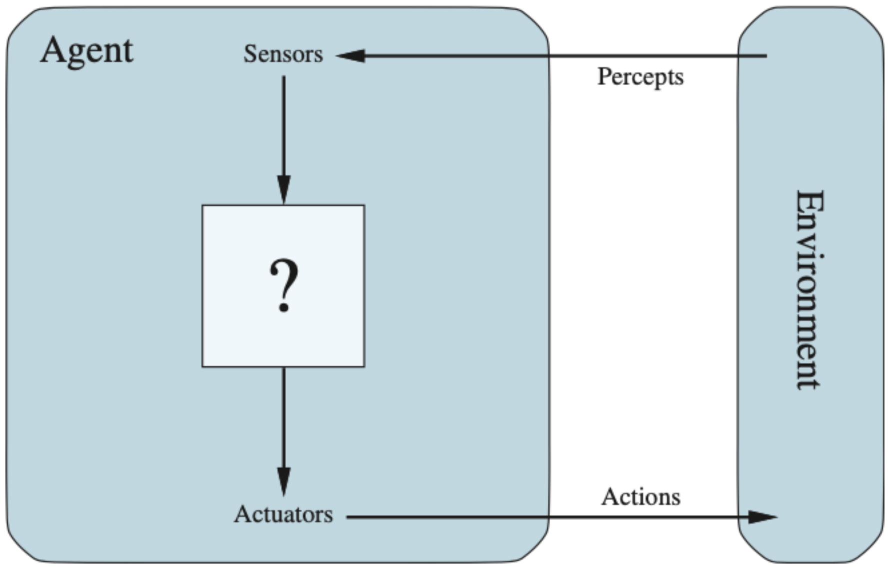


Figure 2.1 Agents interact with environments through sensors and actuators.

What is an (Intelligent) Agent?

» PAGE (Percepts, Actions, Goals, Environment).

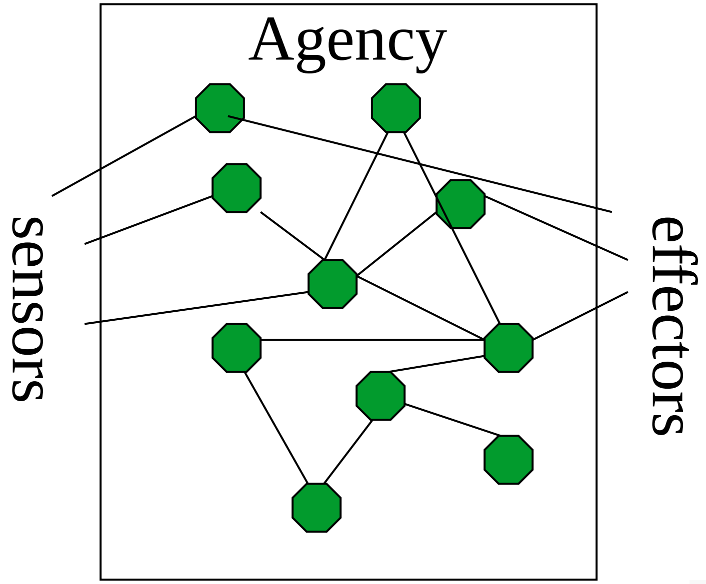
» Task-specific & specialized: well-defined goals and environment.

» The notion of an agent is meant to be a tool for analyzing systems

- Not an absolute characterization that divides the world into agents and non-agents.
- Much like, e.g., object-oriented vs. imperative program design approaches.

Intelligent Agents and AI

- » Human mind as network of thousands or millions of agents all working in parallel.
  - To produce real artificial intelligence, this school holds, we should build computer systems that also contain many agents and systems for arbitrating among the agents' competing results.
- » Distributed decision-making and control
- » **Challenges:**
  - Action selection: What next action to choose
  - Conflict resolution



Example: A Windshield Wiper Agent

How do we design an agent that can wipe the windshields when needed?

» Goals ?

» Percepts ?

» Sensors ?

» Effectors/Actuators?

» Actions ?

» Environment ?

Example: A Windshield Wiper Agent

» **Goals:** To keep windshields clean and maintain good visibility

» **Percepts:** Raining, Dirty

» **Sensors:** Camera (moist sensor)

» **Effectors:** Wipers (left, right, back)

» **Actions:** Off, Slow, Medium, Fast

» **Environment:** US inner city, freeways, highways, weather …

Example: Autonomous Vehicles

Collision Avoidance Agent (CAA)

- Goals: Avoid running into obstacles
- Percepts ?
- Sensors?
- Effectors ?
- Actions ?
- Environment: Freeway

Lane Keeping Agent (LKA)

- Goals: Stay in current lane
- Percepts ?
- Sensors?
- Effectors ?
- Actions ?
- Environment: Freeway

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.

Behavior and Performance of IAs

» Perception (sequence) to Action Mapping:

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

- Ideal mapping: specifies which actions an agent ought to take at any point in time
- Description: Look-Up-Table vs. Closed Form

» 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 actions on its own?

Agent PEAS Description

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments	Touchscreen/voice entry of symptoms and findings
Satellite image analysis system	Correct categorization of objects, terrain	Orbiting satellite, downlink, weather	Display of scene categorization	High-resolution digital camera
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, tactile and joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, raw materials, operators	Valves, pumps, heaters, stirrers, displays	Temperature, pressure, flow, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, feedback, speech	Keyboard entry, voice

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 ?

» Rational = Optimal ?

» Rational = Omniscience ?

» Rational = Clairvoyant ?

» Rational = Successful ?

The Right Thing = The Rational Action

» Rational = Best (Yes, to the best of its knowledge)

» Rational = Optimal (Yes, to the best of its abilities, including its constraints)

» Rational ≠ Omniscience

» Rational ≠ Clairvoyant

» Rational ≠ Successful

How is an Agent different from other software?

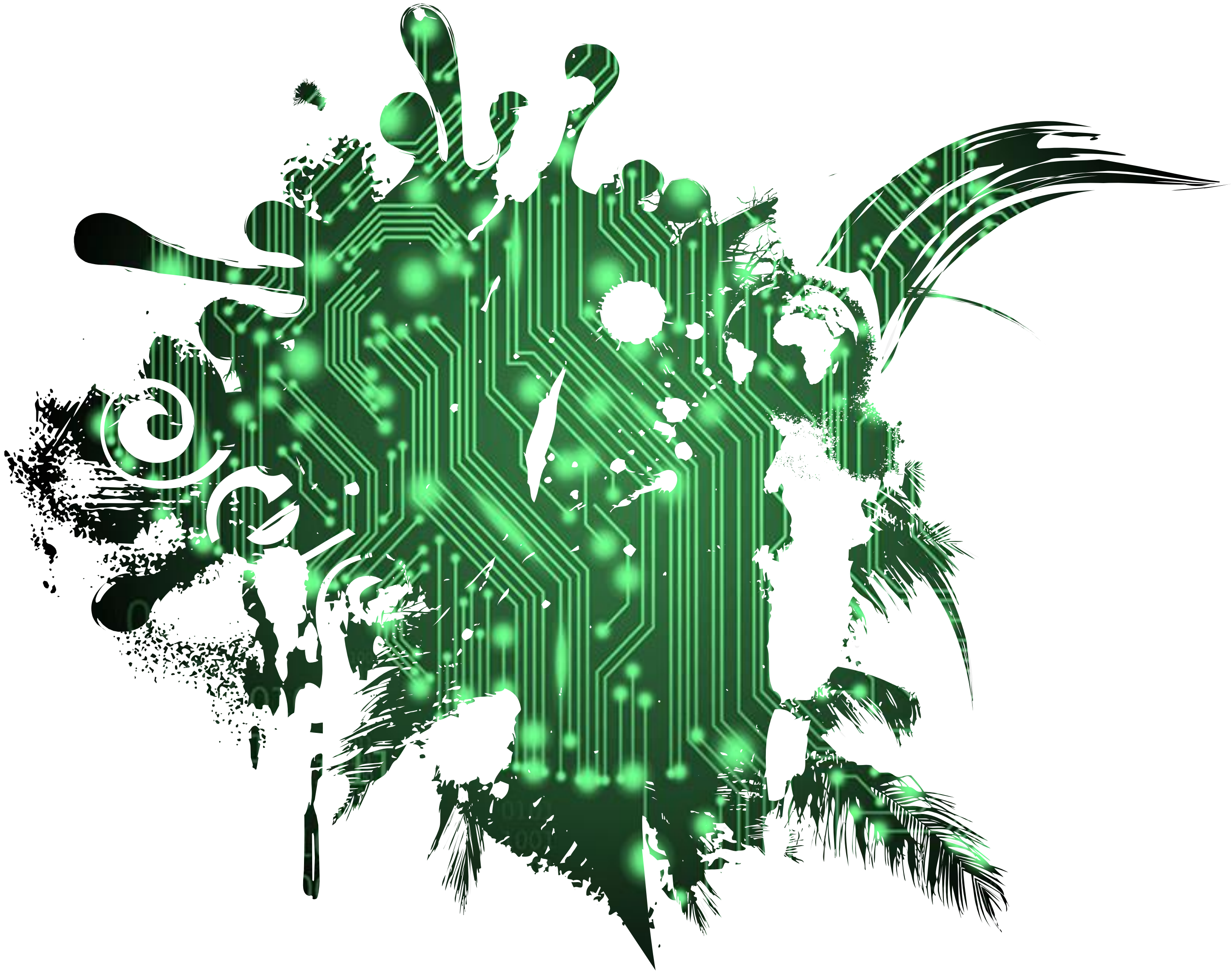
- » Agents are autonomous, that is they act on behalf of the user
- » Agents contain some level of intelligence, from fixed rules to learning engines that allow them to adapt to changes in the environment
- » Agents don't only act reactively, but sometimes also proactively
- » Agents have social ability, that is they communicate with the user, the system, and other agents as required

How is an Agent different from other software?

» 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

Environment Types



Environment Types

Characteristics

» Deterministic vs. nondeterministic

» Episodic vs. non-episodic

» Static vs. dynamic

» Discrete vs. continuous

» Single vs. multi agent

Environment Types

» **Deterministic vs. nondeterministic**

- If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise, it is stochastic.

» **Static vs. dynamic**

- If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise, it is static.

Environment Types

» Episodic vs. non-episodic

- In an episodic task environment, the agent's experience is divided into atomic episodes. In each episode the agent receives a percept and then performs a single action.

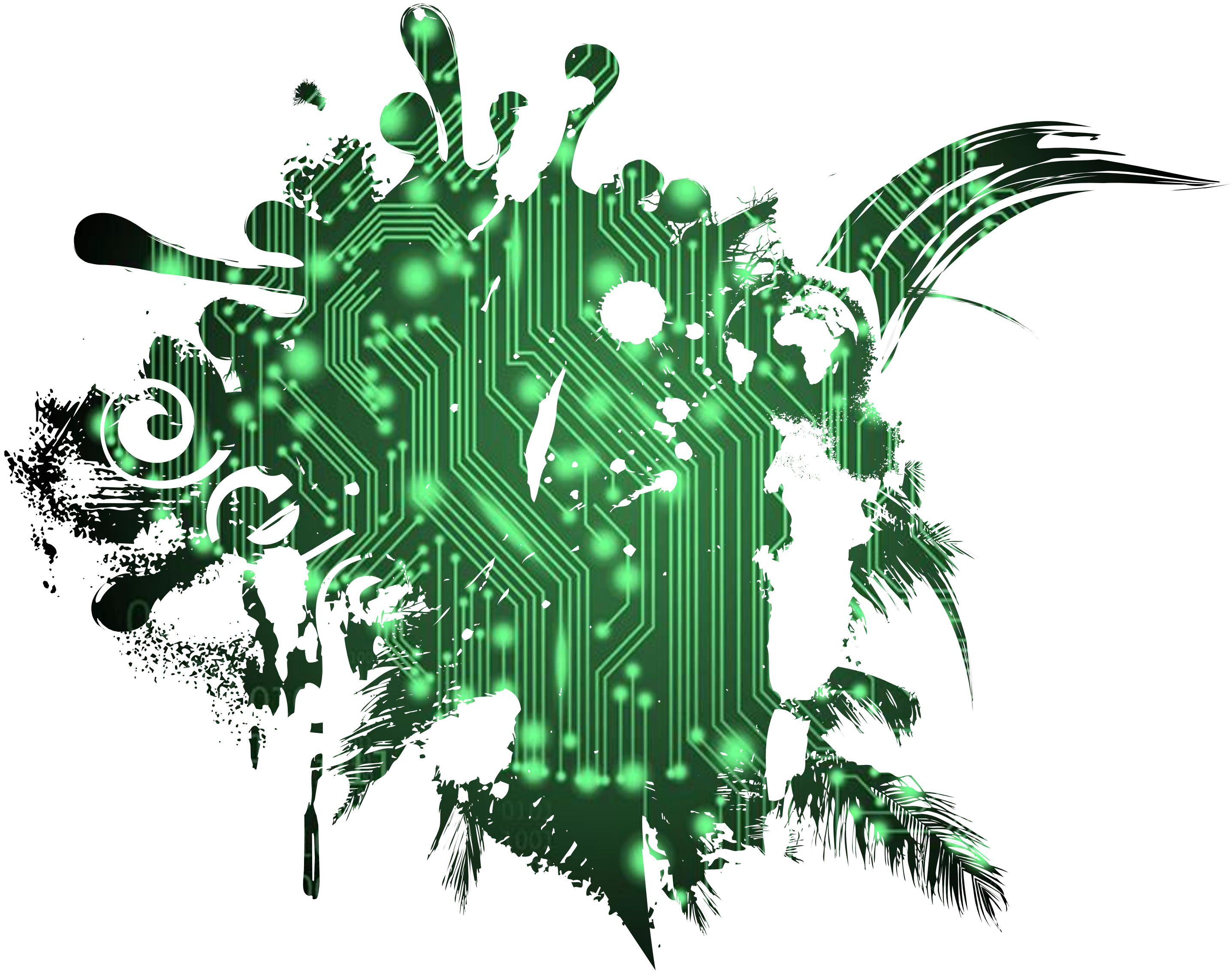
» Discrete vs. continuous

- The discrete/continuous distinction applies to the *state* of the environment, to the way *time* is handled, and to the *percepts* and *actions* of the agent.

Environments

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

IA Structure



Structure of Intelligent Agents

» Agent = architecture + program

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

function: Skeleton-Agent(*Percept*) returns *Action*
memory $\leftarrow$ UpdateMemory(memory, *Percept*)
Action $\leftarrow$ ChooseBestAction(memory)
memory $\leftarrow$ UpdateMemory(memory, *Action*)
return *Action*

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

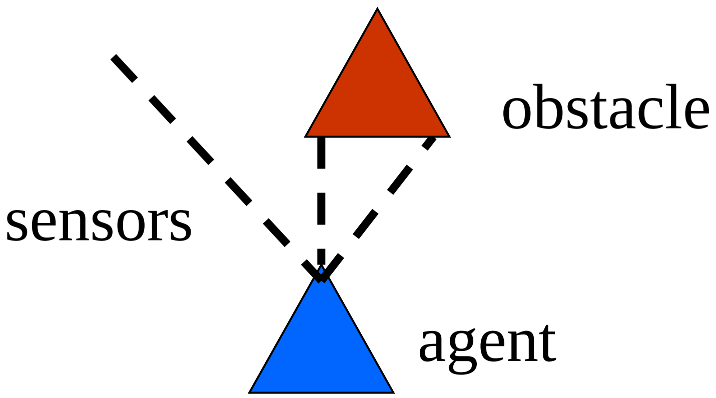
Using a look-up-table

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

» How to generate?

» How large?

» How to select action?

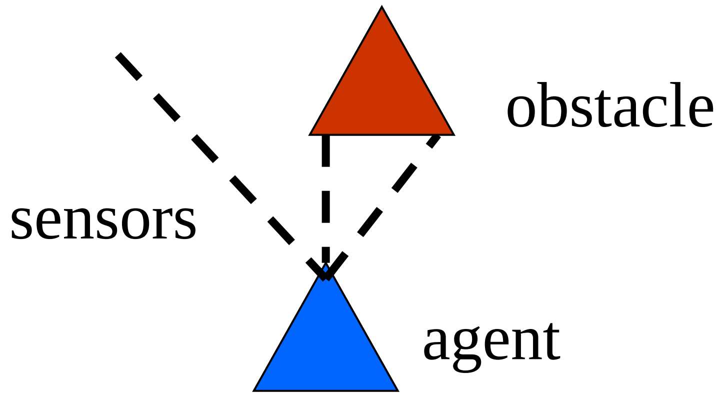


Using a look-up-table

» Example: Collision Avoidance

- Sensors: 3 proximity sensors
- Effectors: Steering Wheel, Brakes

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



Using a look-up-table

» How large: size of table
- = #possible percepts × # possible actions
- = $|\mathcal{P}_\ell|\,|\mathcal{P}_m|\,|\mathcal{P}_r|\,|\mathcal{S}|\,|\mathcal{B}|$

» E.g., P = {close, medium, far}^{3}

» A = {left, straight, right} × {on, off}
- then size of table = 27*3*2 = 162

» How to select action? Search.

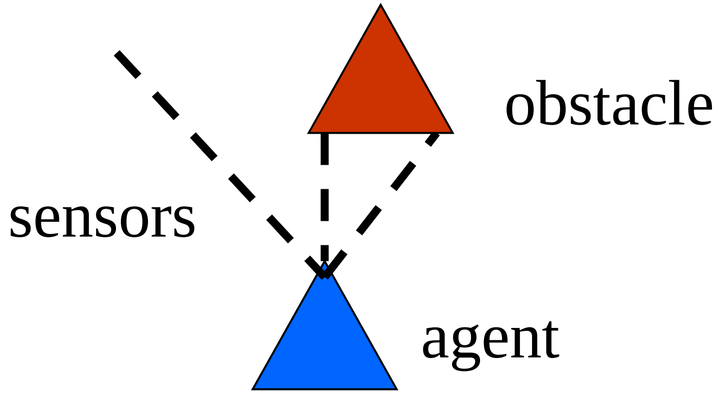


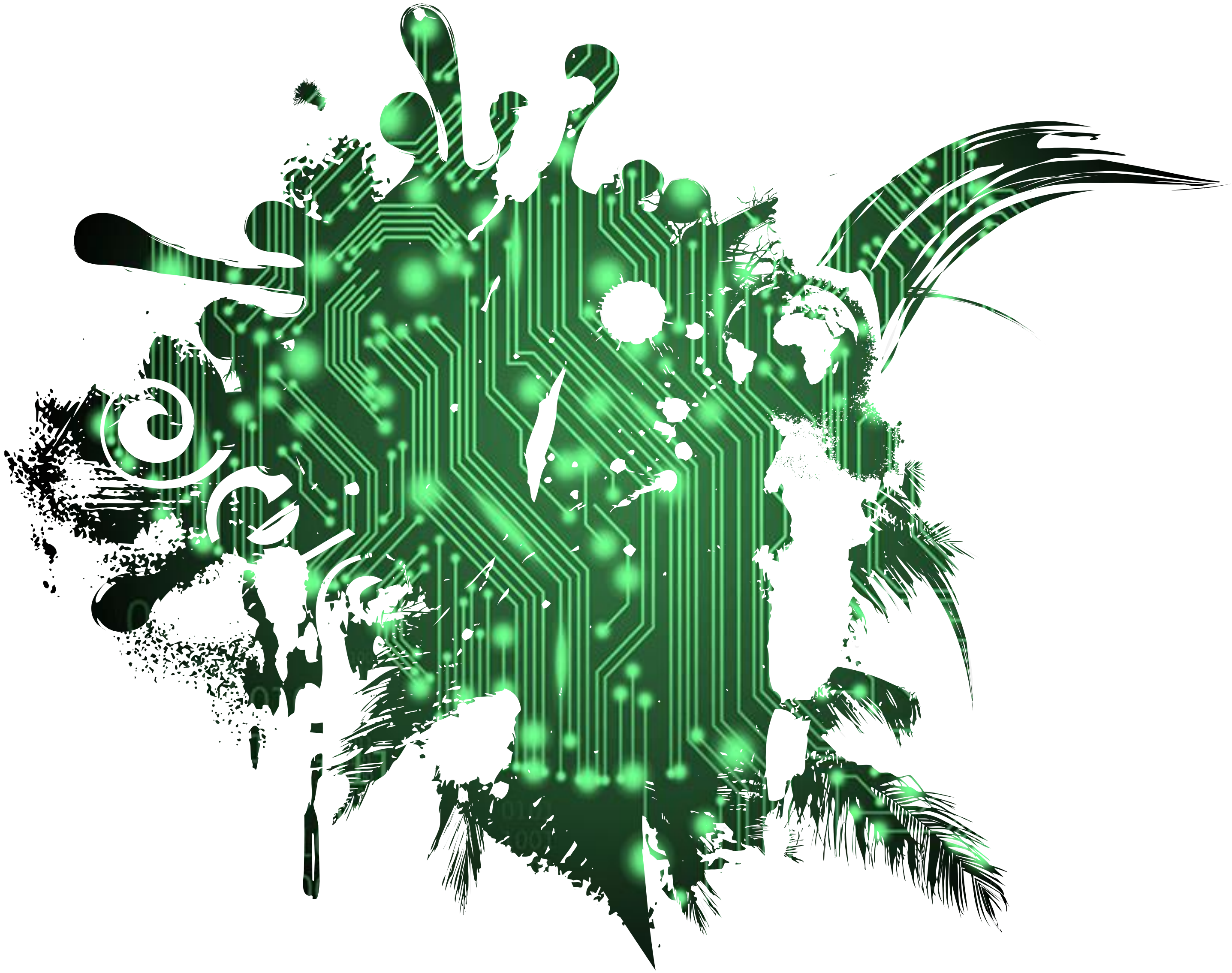
Table-Driven Agent

function TABLE-DRIVEN-AGENT(*percept*) **returns** an action
  persistent: *percepts*, a sequence, initially empty
    table, a table of actions, indexed by percept sequences, initially fully specified

  append *percept* to the end of *percepts*
  action $\leftarrow$ LOOKUP(*percepts*, *table*)
  return *action*

Figure 2.7 The TABLE-DRIVEN-AGENT program is invoked for each new percept and returns an action each time. It retains the complete percept sequence in memory.

IA Types



IA Types

» Simple reflex agents
» Model-based reflex agents
» Goal-based agents
» Utility-based agents
» Learning agents
» Mobile agents

Simple Reflex Agents

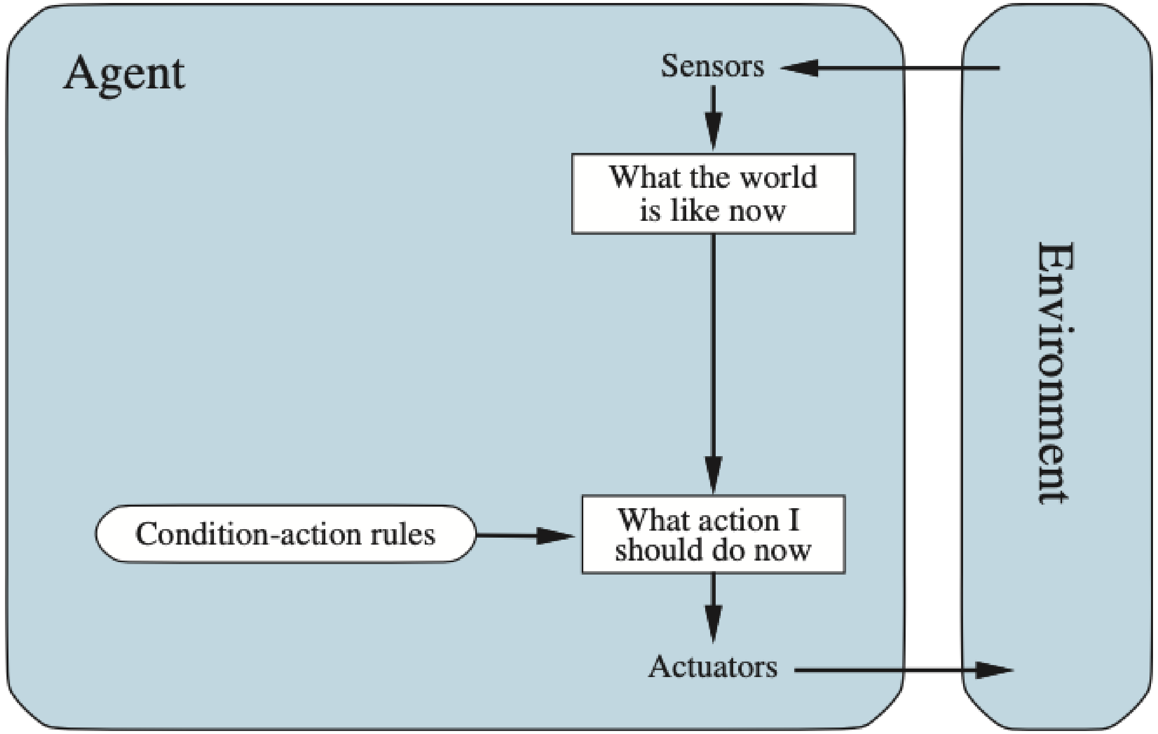


Figure 2.9 Schematic diagram of a simple reflex agent. We use rectangles to denote the current internal state of the agent's decision process, and ovals to represent the background information used in the process.

Simple Reflex 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?
- and how do you debug them?

Simple Reflex Agents

```
function SIMPLE-REFLEX-AGENT(percept) returns an action
  persistent: rules, a set of condition–action rules

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

Figure 2.10 A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

Model-based Reflex Agents

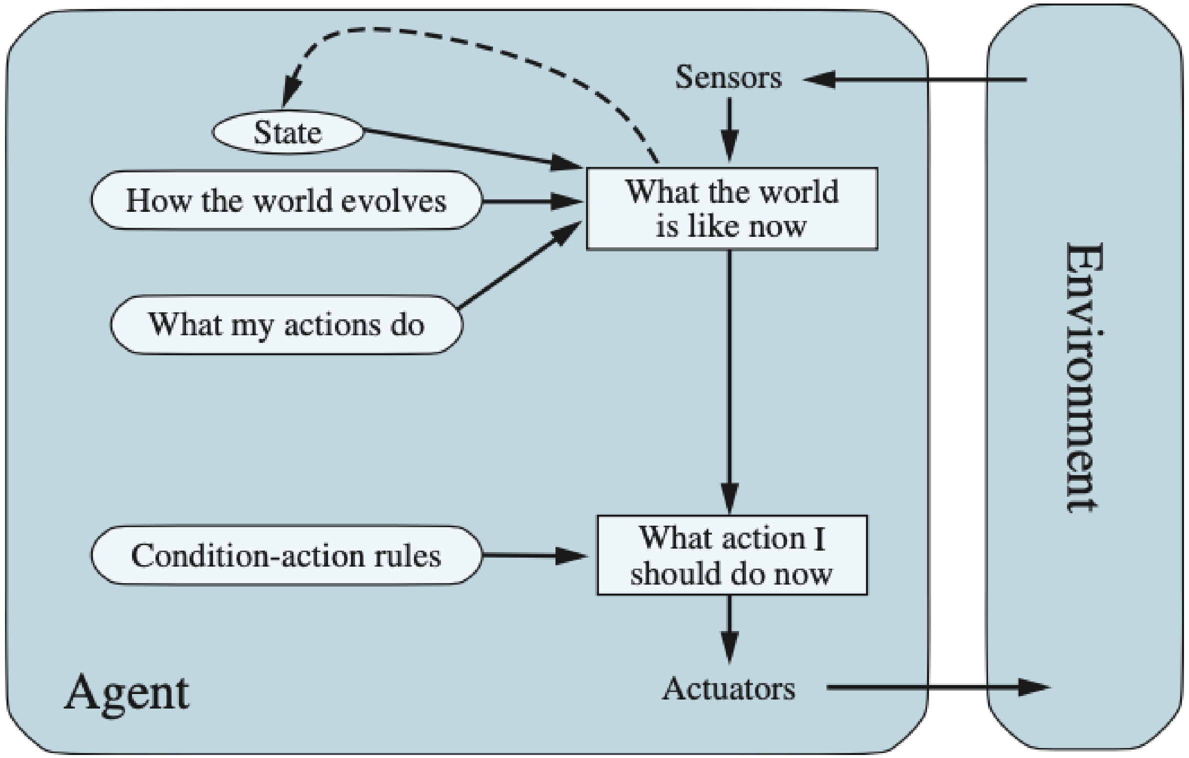


Figure 2.11 A model-based reflex agent.

Model-based Reflex Agents

function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *state*, the agent's current conception of the world state
    transition_model, a description of how the next state depends on
      the current state and action
    sensor_model, a description of how the current world state is reflected
      in the agent's percepts
    rules, a set of condition–action rules
    action, the most recent action, initially none

  state ← UPDATE-STATE(*state*, *action*, *percept*, *transition_model*, *sensor_model*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← *rule*.ACTION
  return *action*

Figure 2.12 A model-based reflex agent. It keeps track of the current state of the world, using an internal model. It then chooses an action in the same way as the reflex agent.

Goal-based Agents

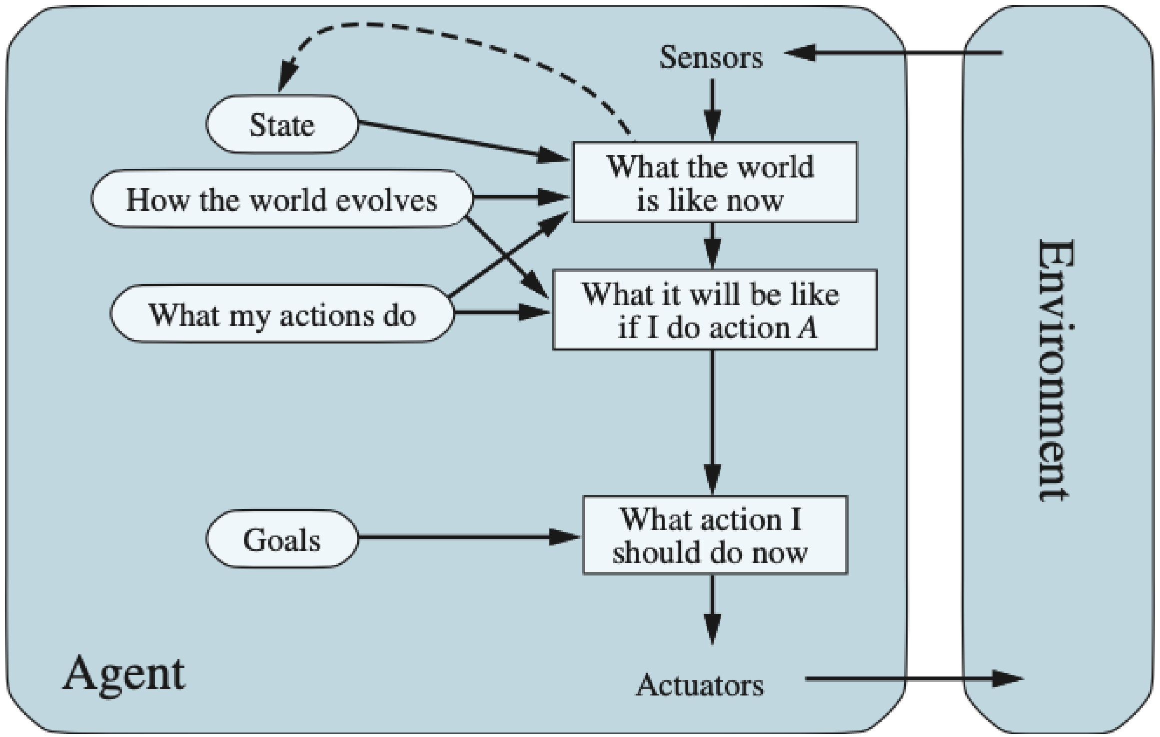


Figure 2.13 A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

Goal-based Agents

» A model-based becomes a goal-based agent, ***if it keeps track of the world state as well as a set of goals it is trying to achieve and chooses an action that will (eventually) lead to the achievement of its goals.***

Utility-based Agents

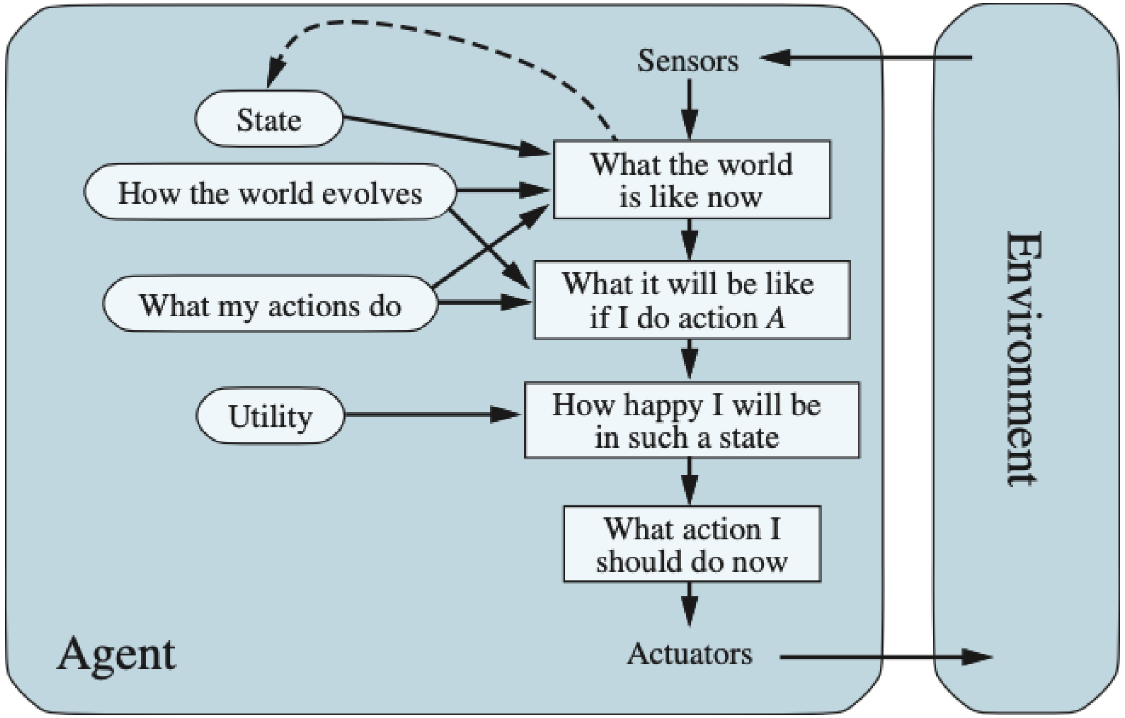


Figure 2.14 A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.

Utility-based Agents

» A model-based becomes a utility-based agent, if

- it uses a model of the world, along with a utility function that measures its preferences among states of the world, then
- it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.

Learning Agents

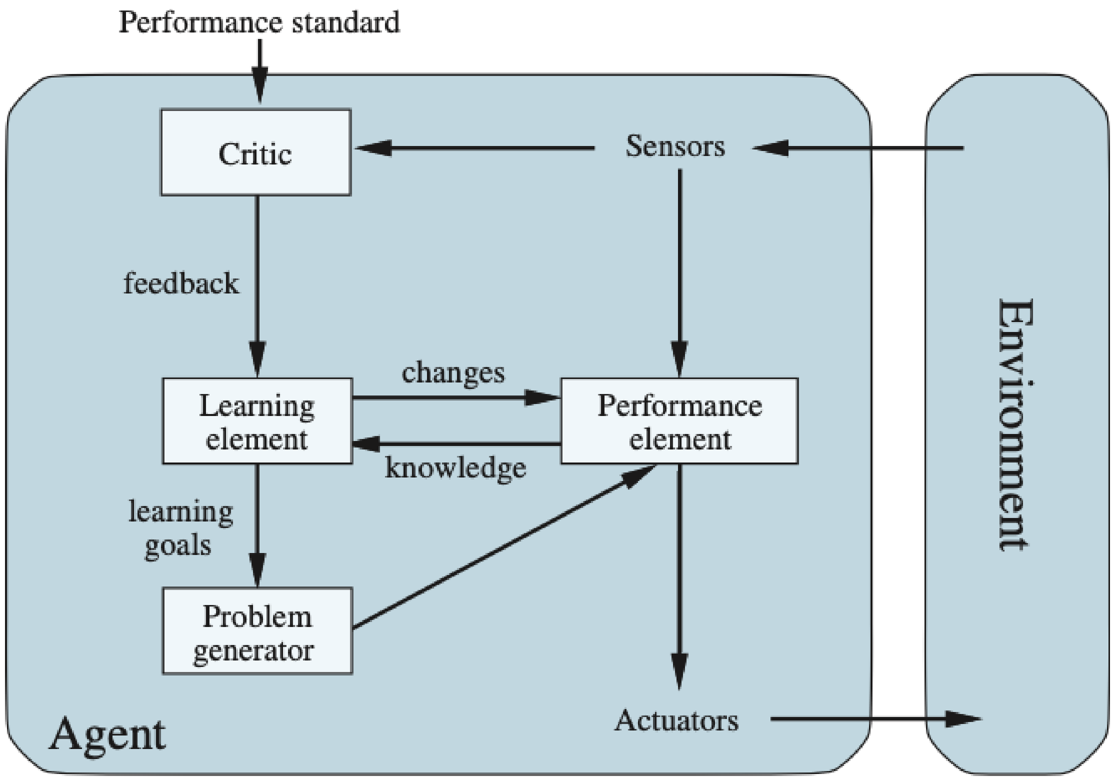


Figure 2.15 A general learning agent. The "performance element" box represents what we have previously considered to be the whole agent program. Now, the "learning element" box gets to modify that program to improve its performance.

Mobile agents

» Programs that can migrate from one machine to another.

» Execute in a platform-independent execution environment.

» Require agent execution environment (places).

» Mobility not necessary or sufficient condition for agenthood.

» Practical but non-functional advantages:

- Reduced communication cost
- Asynchronous computing (when you are not connected)

Conclusion

- **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.
- **Agent Types:** Reflex, Model-based, Goal-based, Utility-based, Learning
- **Rational Action:** The action that maximizes the expected value of the performance measure given the percept sequence to date

Questions & Comments