Intelligent Agents

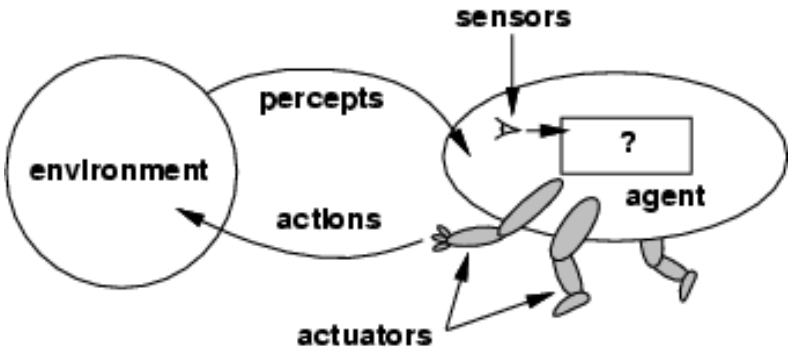
Outline

- Agents and environments
- Rationality
- PEAS (Performance measure, Environment, Actuators, Sensors)
- Environment types
- Agent types

Agents

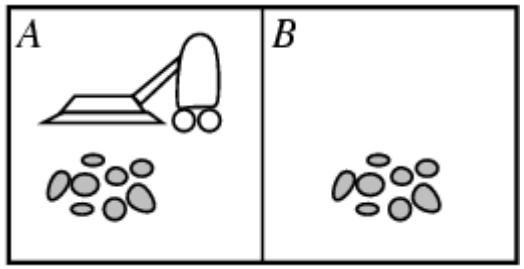
- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators
-
- Human agent: eyes, ears, and other organs for sensors; hands,
- legs, mouth, and other body parts for actuators
-
- Robotic agent: cameras and infrared range finders for sensors;
- various motors for actuators
-

Agents and environments



- The agent function maps from percept histories to actions:
- $[f: P^* \rightarrow A]$
- The agent program runs on the physical architecture to produce *f*
-

Vacuum-cleaner world



- Percepts: location and contents, e.g., [A,Dirty]
-
- Actions: *Left*, *Right*, *Suck*, *NoOp*
-

Rational agents

- An agent should strive to "do the right thing", based on what it can perceive and the actions it can perform. The right action is the one that will cause the agent to be most successful
-
- Performance measure: An objective criterion for success of an agent's behavior
-
- E.g., performance measure of a vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.
-

Rational agents

- 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.
-

Rational agents

- Rationality is distinct from omniscience (all-knowing with infinite knowledge)
-
- Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration)
-
- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt)

PEAS

- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent design
-
- Consider, e.g., the task of designing an automated taxi driver:
-
  - Performance measure
  -
  - Environment
  - Actuators
  - Sensors
  -

PEAS

- Must first specify the setting for intelligent agent design
-
- Consider, e.g., the task of designing an automated taxi driver:
-
  - Performance measure: Safe, fast, legal, comfortable trip, maximize profits
  -
  - Environment: Roads, other traffic, pedestrians, customers
  -
  - Actuators: Steering wheel, accelerator, brake, signal, horn

PEAS

- Agent: Medical diagnosis system
- Performance measure: Healthy patient, minimize costs, lawsuits
- Environment: Patient, hospital, staff
- Actuators: Screen display (questions, tests, diagnoses, treatments, referrals)
-
- Sensors: Keyboard (entry of symptoms, findings, patient's answers)

PEAS

- Agent: Part-picking robot
- Performance measure: Percentage of parts in correct bins
- Environment: Conveyor belt with parts, bins
- Actuators: Jointed arm and hand
- Sensors: Camera, joint angle sensors

PEAS

- Agent: Interactive English tutor
- Performance measure: Maximize student's score on test
- Environment: Set of students
- Actuators: Screen display (exercises, suggestions, corrections)
- Sensors: Keyboard

Environment types

- Fully observable (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.
-
- Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic)
-
- Episodic (vs. sequential): The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.
-

Environment types

- Static (vs. dynamic): The environment is unchanged while an agent is deliberating. (The environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does)
-
- Discrete (vs. continuous): A limited number of distinct, clearly defined percepts and actions.
-
- Single agent (vs. multiagent): An agent operating by itself in an environment.
-

Environment types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

- The environment type largely determines the agent design
-
- The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent
-

Properties of a Chess agent.

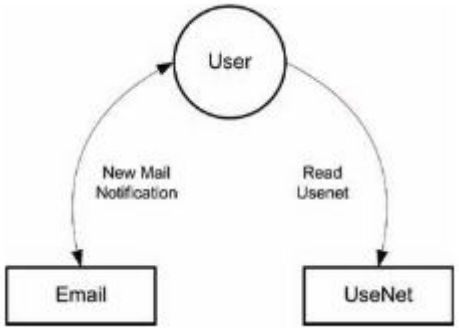
Property	Description
Fully Observable	The Chess board is fully observable to the Chess agent, nothing is hidden.
Static	The environment changes based on actions of the Chess agent and those of the opponent. But during the period when the Chess agent is making a decision for a move, the environment (Chess board) does not change.
Determistic	The Chess board changes based on the move selected by the agent, and therefore the environment is deterministic.
Episodic	The Chess agent operates in episodes, alternating between agent moves and opponent moves.
Multi-Agent	The Chess board environment can be classified as single agent (if the opponent is not considered) or as multi-agent, considering that an opponent operates on the environment in a competitive fashion

Properties of a non-player character agent.

Property	Description
Partial Observability	Another character in the game may not be visible to the player,
Dynamic	Players and NPCs compete or cooperate in the environment i n real-time,
Stochastic	An action taken by an agent at one time may not result in the same response when taken again (such as shooting at another player).
Continuous	The FPS environment is continuous, as compared to an episodic environment such as a turn-based strategy game.
Multi-Agent	Typically, these are competitive environments, though some also include cooperative elements through support NPC agents.

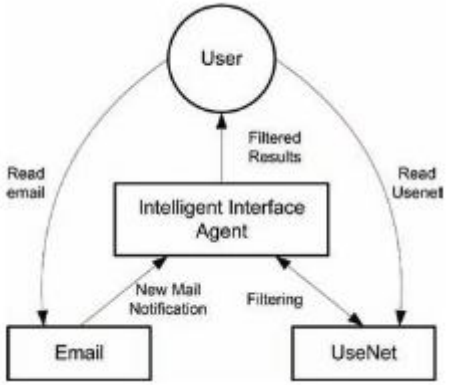
AGENT TAXONOMIES

- Interface Agents
  - To minimize information overload on a user, and reduce the amount of information presented to a user as a way to help the user focus on what is most important at any given time



A user scenario for email and UseNet.

- Virtual Character Agents
  - A useful agent application that take on a number of forms



An intelligent interface agent to minimize distractions.

AGENT TAXONOMIES

- Entertainment Agents
    - used as characters in computer-generated (CG) movies or for training purposes in military simulations.
- Game Agents
    - Non-Player Characters in games (NPCs), bring life to a variety of games by introducing characters that are autonomous and add to the realism of video games.
- ChatterBots, or conversational agents
- Mobile Agents
    - the agents have the ability to migrate from one host computer to another.
- User Assistance Agent
    - for the purpose of simplifying our experiences when dealing with computers
    - Examples: Email filtering, Information Gathering and Filtering and other user assistance application
- Hybrid Agents
    - Instead of a single characteristic, such as mobile, agents implement multiple characteristics, such as mobile and communicative.

Agent functions and programs

- An agent is completely specified by the agent function mapping percept sequences to actions
- One agent function (or a small equivalence class) is rational
-
- Aim: find a way to implement the rational agent function concisely
-

Table-lookup agent

- \input{algorithms/table-agent-algorithm}
-
- Drawbacks:
  - Huge table
  - Take a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries

Agent program for a vacuum-cleaner agent

- \input{algorithms/reflex-vacuum-agent-algorithm}
-

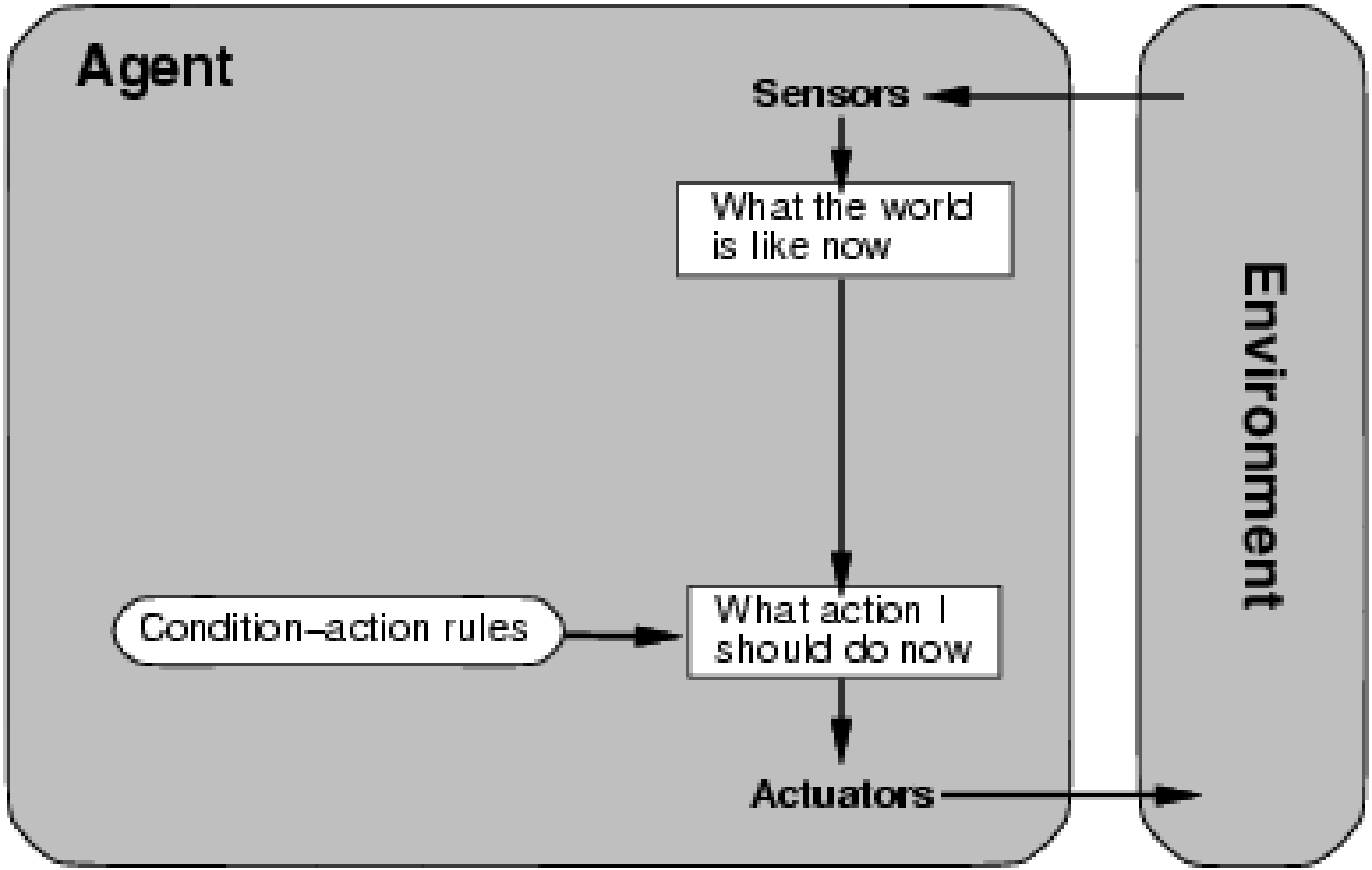
Agent properties.

Property	Description
Rationale	Able to act in a rational (or intelligent) way (does the right thing at the right time, given a known outcome.)
Autonomous	Able to act independently, not subject to external control
Persistent	Able to run continuously
Communicative	Able to provide information, or command other agents
Cooperative	Able to work with other agents to achieve goals
Mobile	Able to move (typically related to network mobility)
Adaptive	Able to learn and adapt

Agent types

- Four basic types in order of increasing generality:
-
- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents

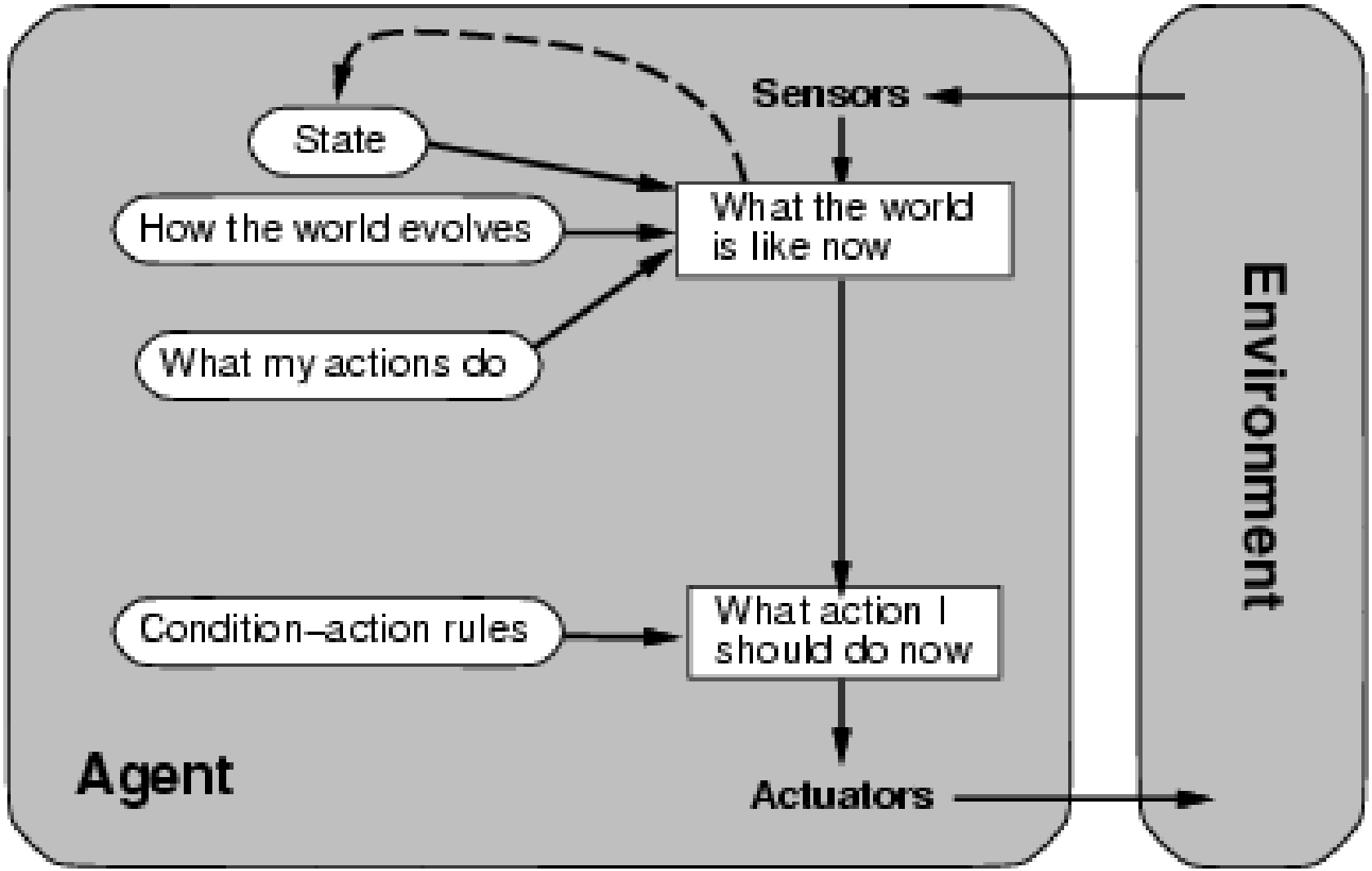
Simple reflex agents



Simple reflex agents

- \input{algorithms/d-agent-algorithm}
-

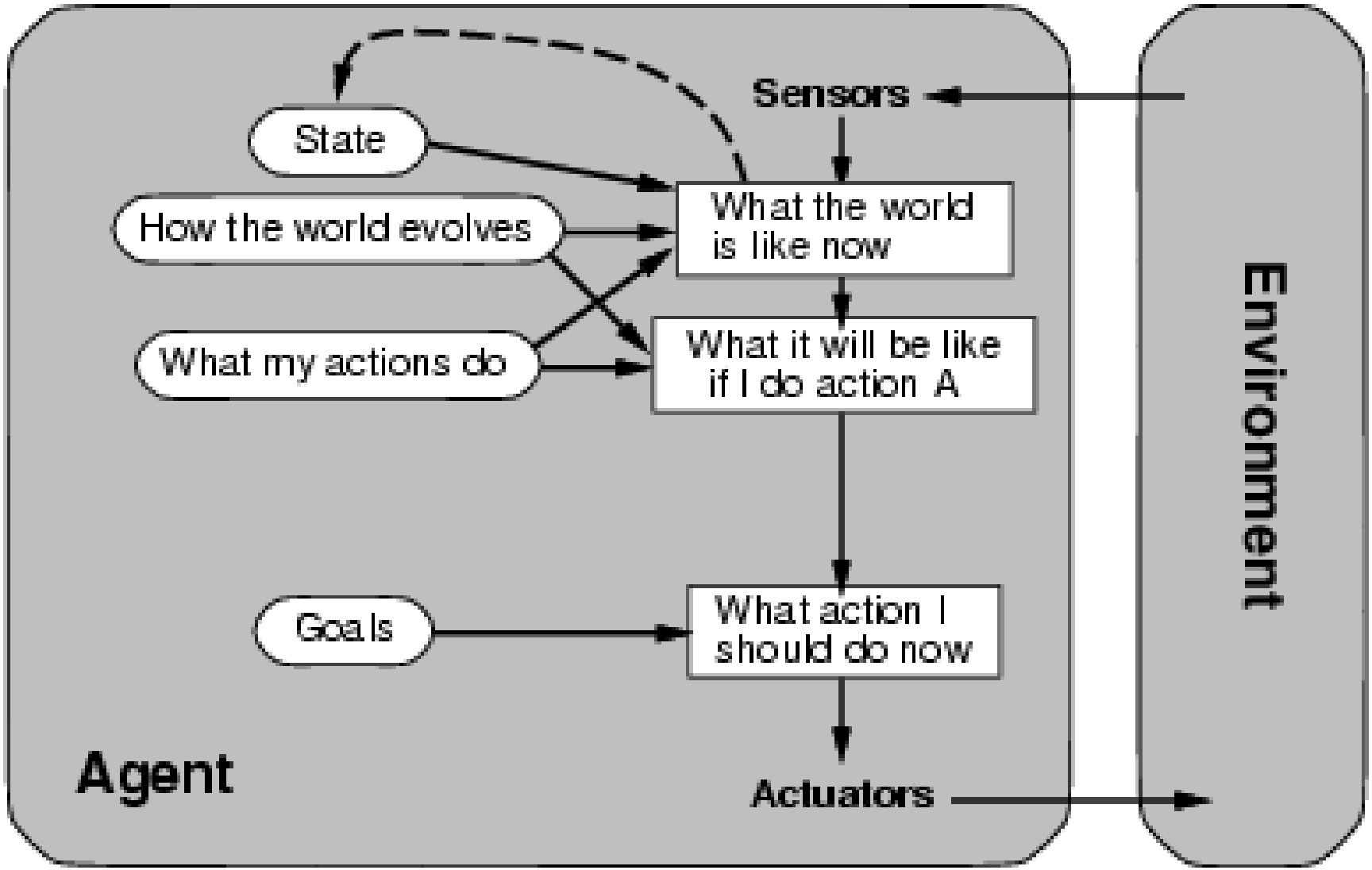
Model-based reflex agents



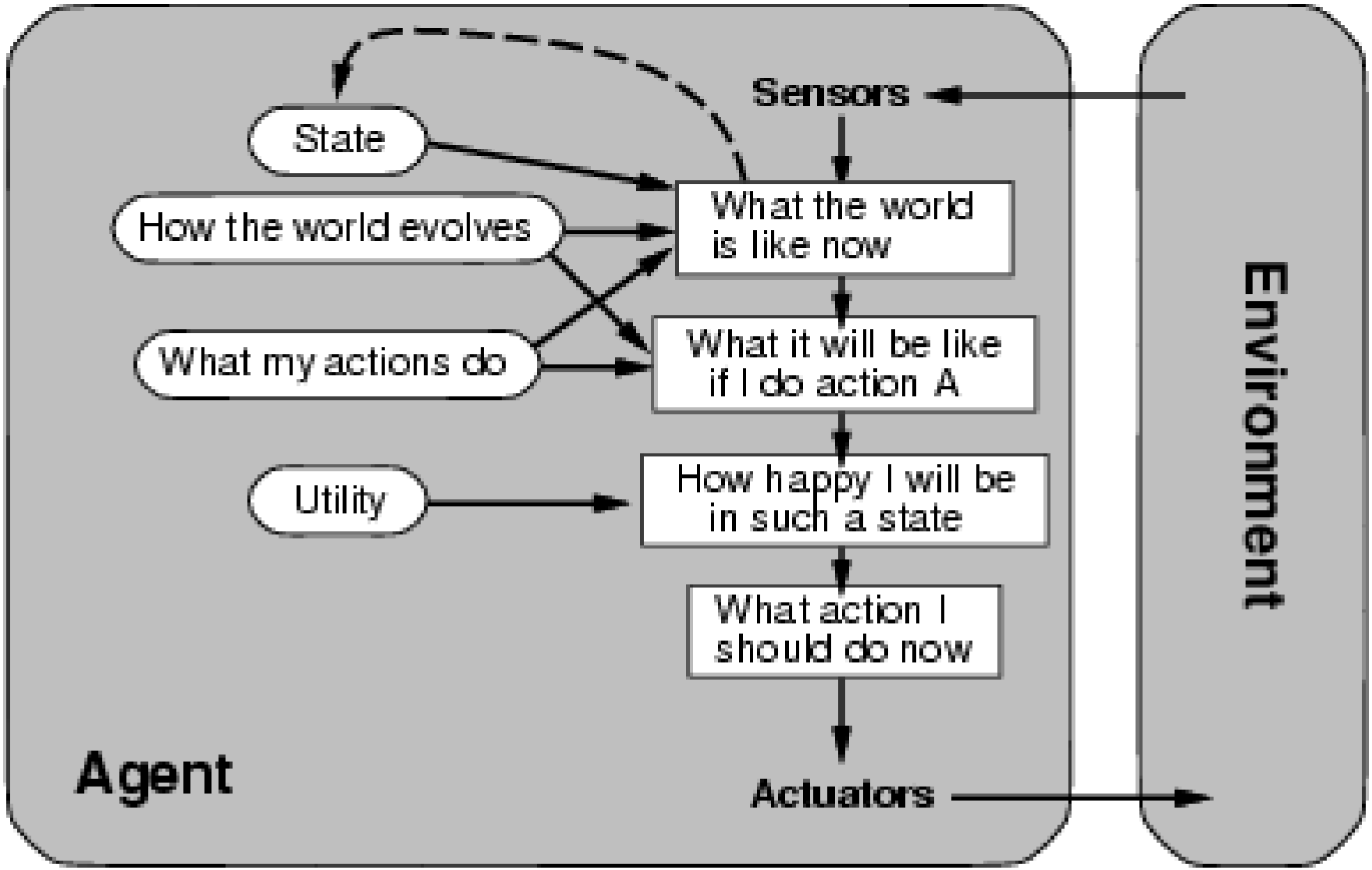
Model-based reflex agents

- \input{algorithms/d+-agent-algorithm}
-

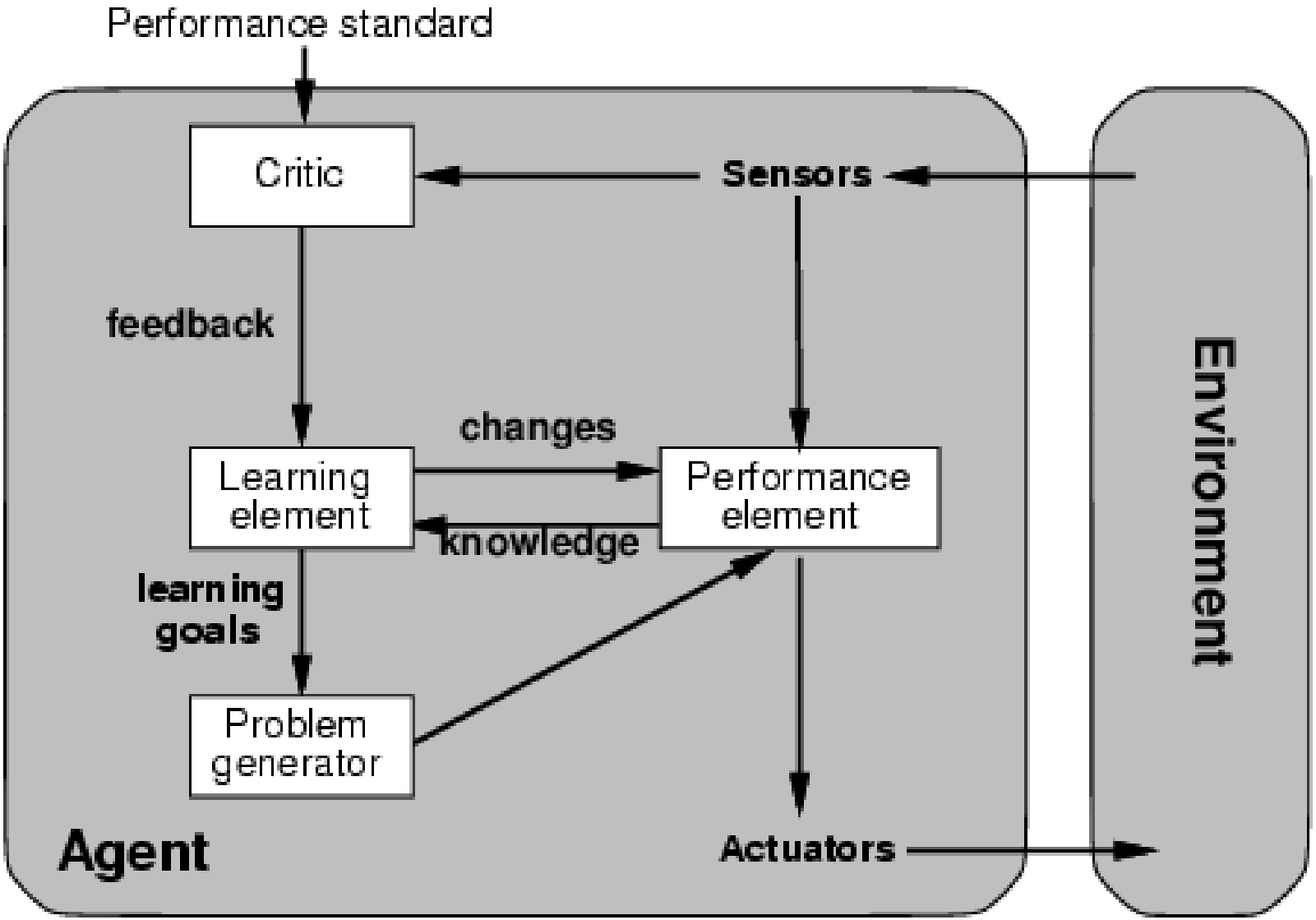
Goal-based agents



Utility-based agents

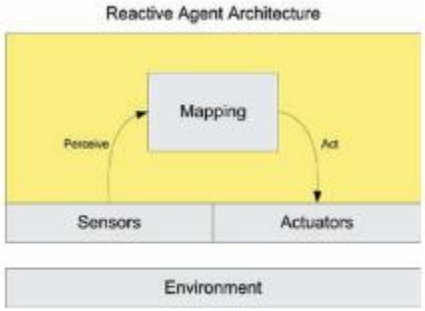


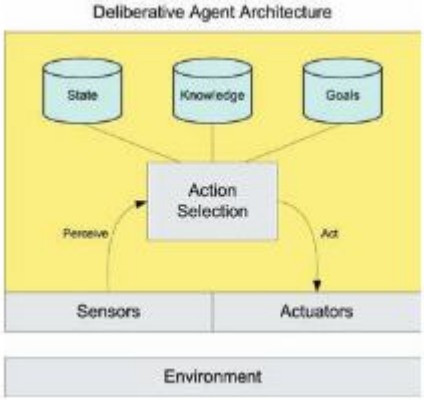
Learning agents



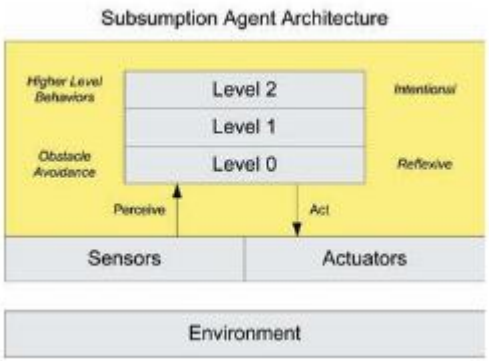
AGENT ARCHITECTURES

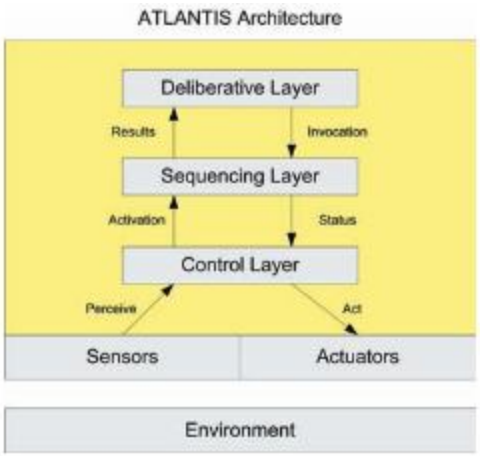
- Reactive/ Reflex Architectures
  - agent behaviors are simply a mapping between stimulus and response.
  - The agent has no decision-making skills, only reactions to the environment in which it exists
  - An example is Subsumption agent
  - Behavior networks, created by Pattie Maes in the late 1980s, is another reactive architecture that is distributed in nature



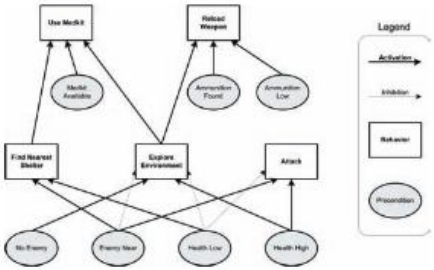


- Deliberative Architectures
  - one that includes some deliberation over the action to perform given the current set of inputs
  - Example is ATLANTIS



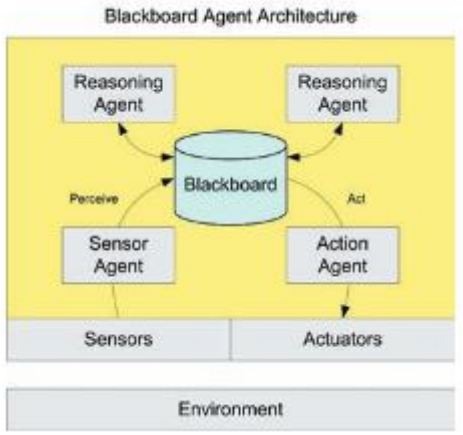


Behaviour network for a simple game agent



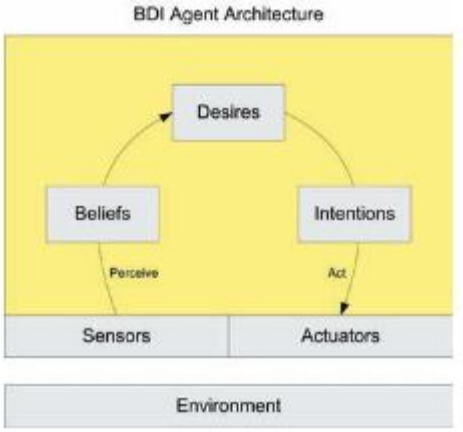
AGENT ARCHITECTURES

- Blackboard Architectures
  - operates around a global work area call the blackboard (a common work area for a number of agents that work cooperatively to solve a given problem



The blackboard architecture supports multi-agent problem solving.

- Belief-Desire-Intention (BDI) Architecture
  - follows the theory of human reasoning as defined by Michael Bratman.
    - Belief represents the view of the world by the agent (what it believes to be the state of the environment in which it exists).
    - Desires are the goals that define the motivation of the agent (what it wants to achieve).
    - Intentions specify that the agent uses the Beliefs and Desires in order to choose one or more actions in order to meet the desires.



AGENT ARCHITECTURES

- Hybrid Architectures
  - Based on the needs of the agent system, different architectural elements can be chosen to meet those needs
- Mobile Architectures
  - introduces the ability for agents to migrate themselves between hosts