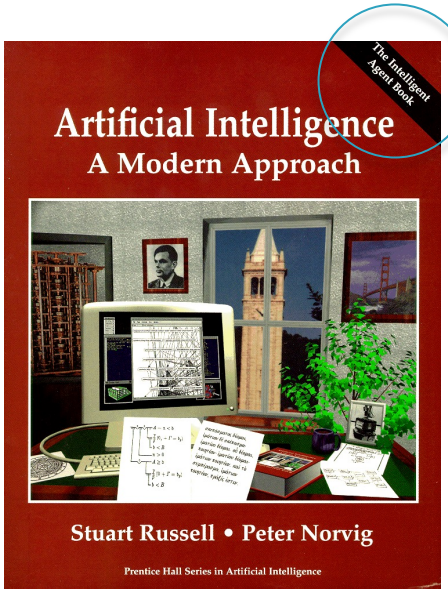
Intelligent Agents

Larry Holder
School of EECS
Washington State University

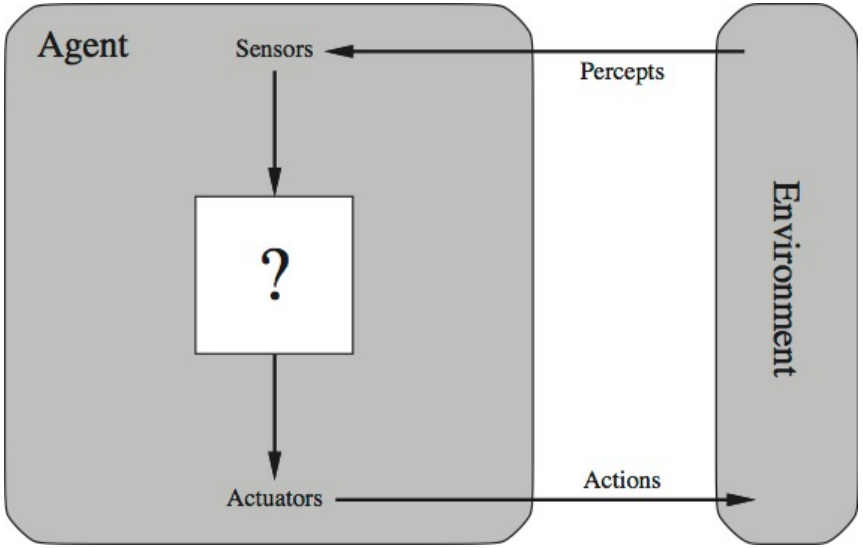
Overview

- What is an agent?
- Rational agent
- Types of environments
- Types of agents



Agent

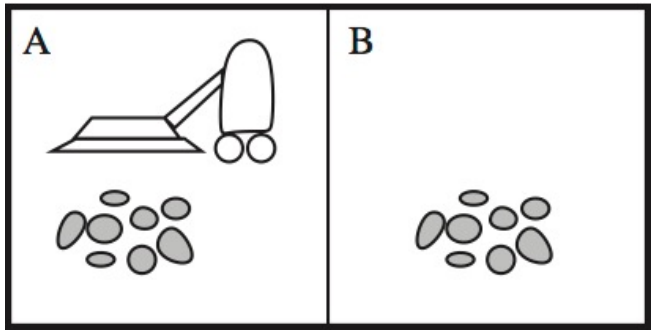
- Agent perceives its *environment* through *sensors* and acts on its environment through *actuators*
- Percepts: Perceptual inputs to the agent
- Percept sequence: complete history of the agent's percepts



Vacuum Agent

- Input: Percept = [location, state]
  - Location ∈ {A, B}
  - State ∈ {Clean, Dirty}
- Return: Action ∈ {Left, Right, Suction}

Vacuum World

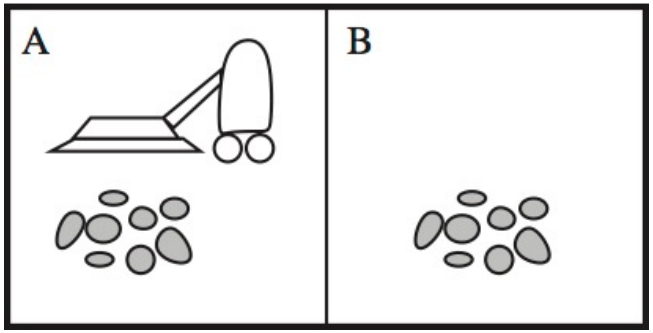


Vacuum Agent

```
Action VacuumAgent (Percept p)
{
   if (p = [?, Dirty])
   then return Suction
   if (p = [A, Clean])
   then return Right
   if (p = [B, Clean])
   then return Left
}
```

Rational Agent

- Rational Agent takes actions that maximize the performance measure given the percept sequence and any prior knowledge
  - Performance measures?
  - Prior knowledge?
- Is VacuumAgent rational?



Self-Driving Car

CES 2019: www.youtube.com/watch?v=gfWjsKsEry0

Rational Agent

- Should a rational agent:
  - Know everything?
  - Explore?
  - Learn?

Task Environment

- PEAS
  - Performance
  - Environment
  - Actuators
  - Sensors

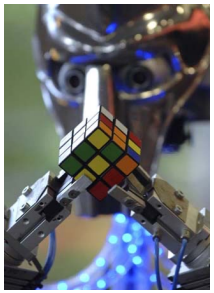
Agent Type	Performance	Environment	Actuators	Sensors
Self-driving car				

Task Environment Examples

Agent Type	Performance	Environment	Actuators	Sensors
Puzzle solver				
Part picker				

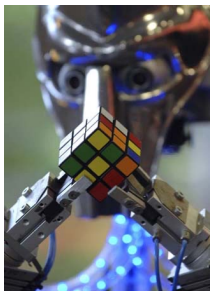
Task Environment Properties

- Fully observable vs. partially observable
  - Do sensors give complete state of environment
- Single agent vs. multiagent
  - Are their other agents in the environment whose performance is affected by this agent
- Puzzle solver?
- Part picker?



Task Environment Properties

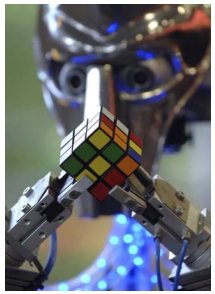
- Deterministic vs. stochastic
  - Next state of environment completely determined by current state and agent's action
- Episodic vs. sequential
  - Future percepts and actions do not depend on past percepts and actions
- Puzzle solver?
- Part picker?



Task Environment Properties

- Static vs. dynamic
  - Can the environment change while the agent is deliberating
- Discrete vs. continuous
  - Are there a fixed number of environment states
- Known vs. unknown
  - Are the effects of actions known

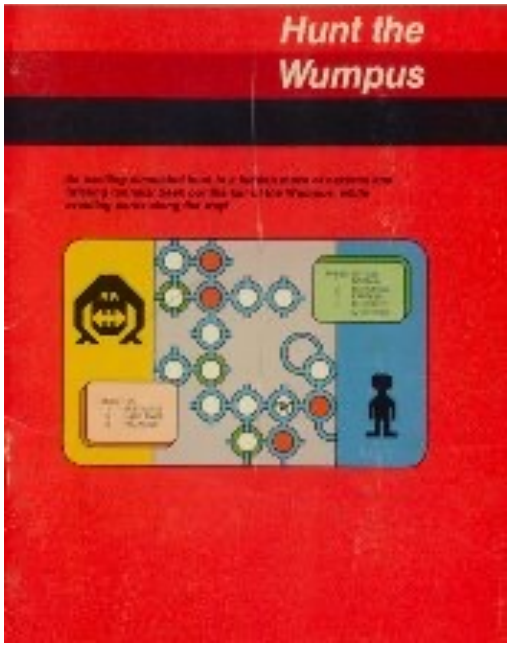
- Puzzle solver?

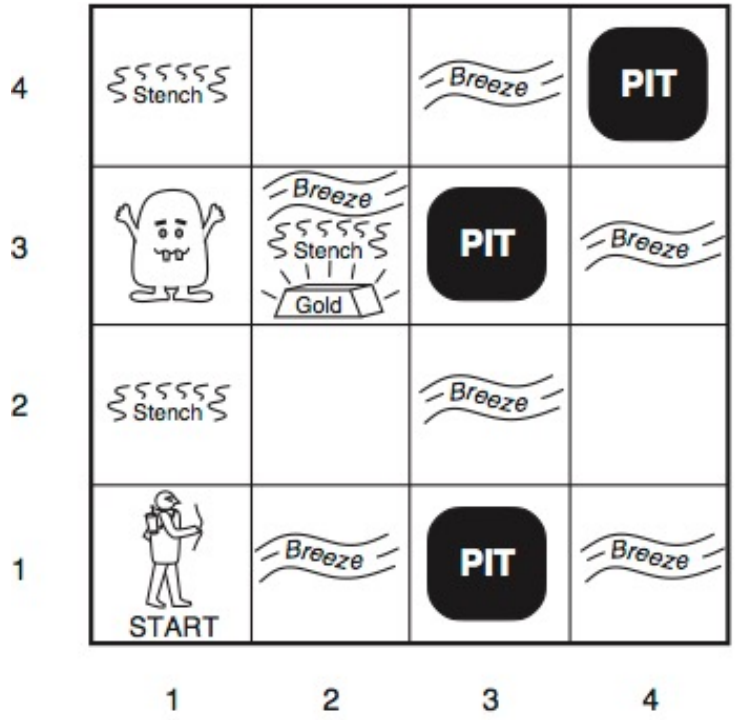


- Part picker?

Wumpus World

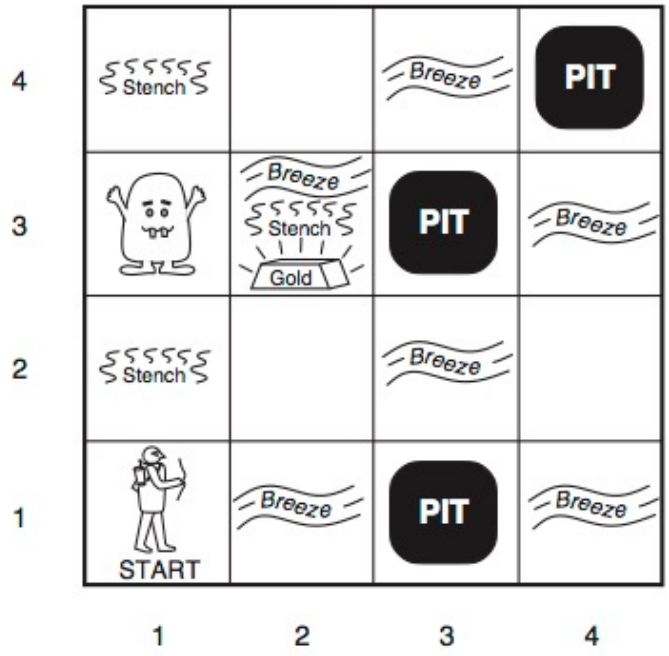
- Hunt the Wumpus game
  - Written in BASIC, 1972
  - First available on the TI-99/4A





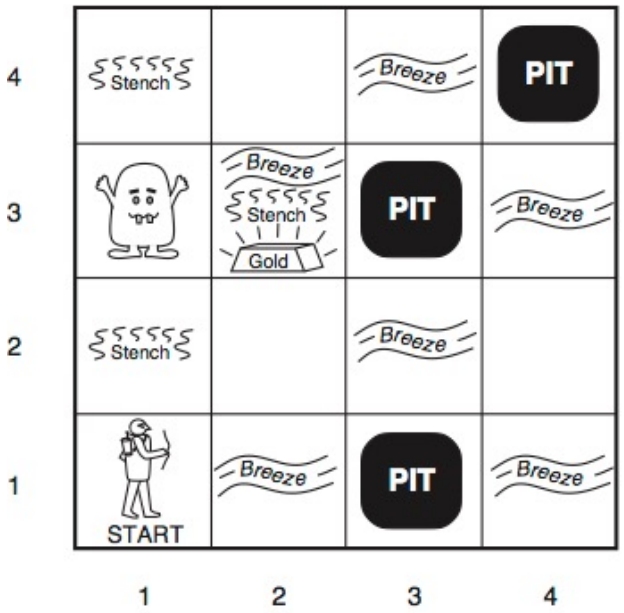
Wumpus World (PEAS)

- Performance measure
  - +1000 for leaving cave with gold
  - -1000 for falling in pit or being eaten by wumpus
  - -1 for each action taken
  - -10 for using the arrow
  - Game ends when agent dies or leaves cave



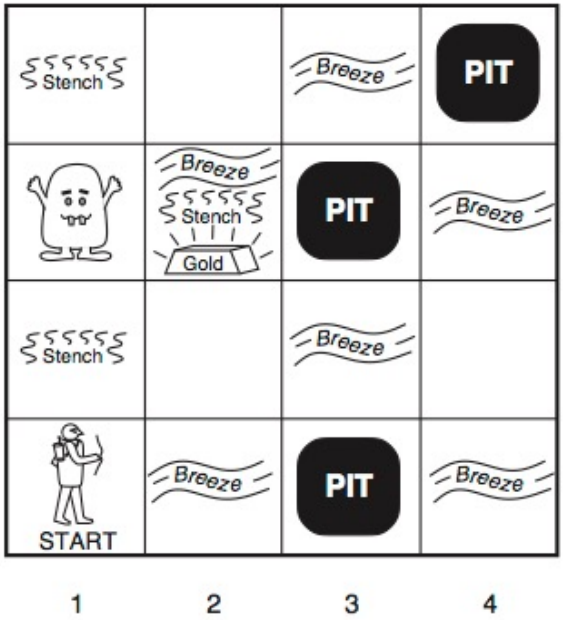
Wumpus World (PEAS)

- Environment
  - 4x4 grid of rooms
  - Agent starts in square [1,1] facing right with 1 arrow
  - Location of wumpus and gold chosen at random other than [1,1]
  - Each square other than [1,1] has a 0.2 probability of containing a pit



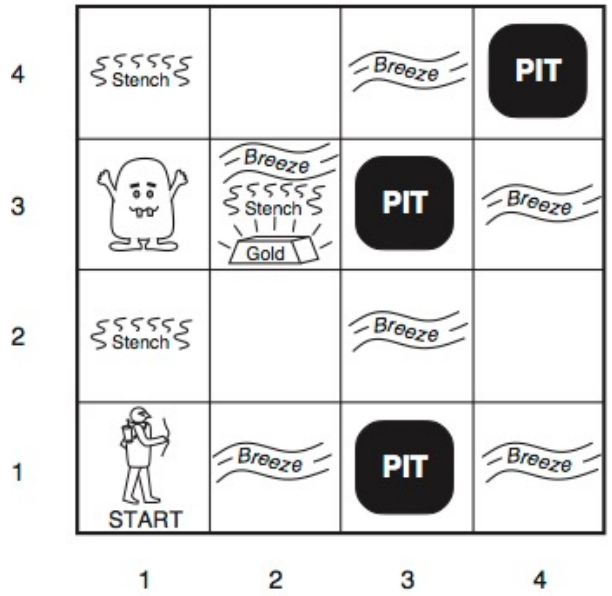
Wumpus World (PEAS)

- Actuators
  - Forward
  - TurnLeft by 90°
  - TurnRight by 90°
  - Grab picks up gold if agent in gold location
  - Shoot shoots arrow in direction agent is facing
    - Arrow continues until hits wumpus or wall
  - Climb leaves cave if agent in [1,1]



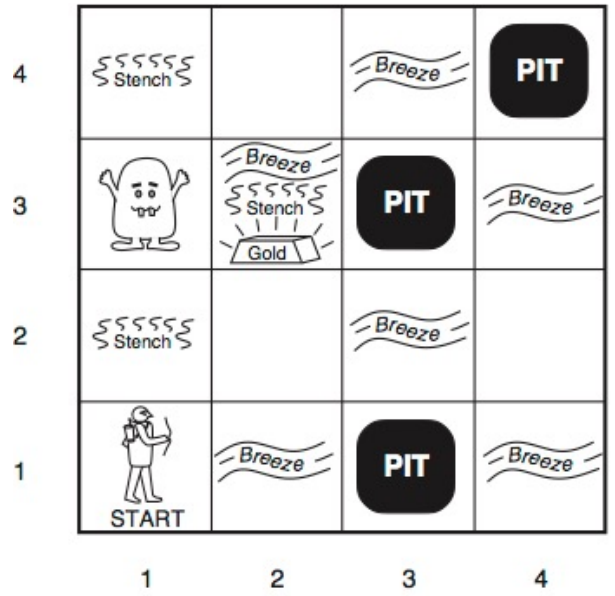
Wumpus World (PEAS)

- Sensors (Boolean)
  - Stench if wumpus in directly (not diagonally) adjacent square
  - Breeze if pit in directly adjacent square
  - Glitter if gold in agent's current square
  - Bump if agent walks into a wall
  - Scream if wumpus is killed



Wumpus Environment

- Fully or partially observable?
- Discrete or continuous?
- Static or dynamic?
- Deterministic or stochastic?
- Single or multi-agent?
- Episodic or sequential?
- Known or unknown?



Basic Agent Program

- Details of design based on task (PEAS) and properties of environment

```
Action Agent (Percept percept)
{
   Process percept
   Choose action
   return action
}
```

Table-driven Agent

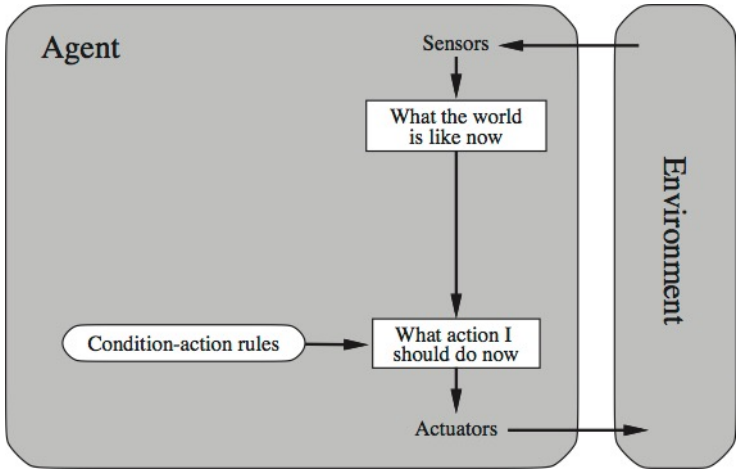
- Table: Percepts → Actions
- Where does table come from?
- How large is table?

```
Action TableDrivenAgent (Percept percept)
{
   PerceptSequence percepts
   Table T

   Append percept to end of percepts
   action = Lookup (percepts, T)
   return action
}
```

Simple Reflex Agent

- Where do rules come from?
- Random component to avoid repetitive behavior

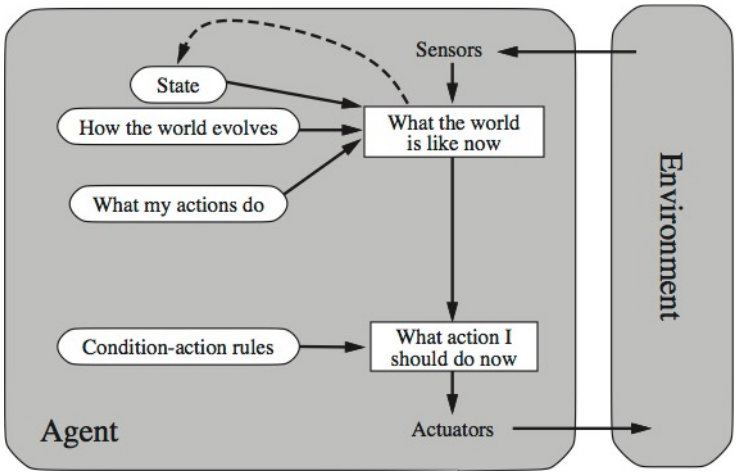



```
Action SimpleReflexAgent (Percept percept)
{
   RuleSet rules

   state = InterpretInput (percept)
   rule = RuleMatch (state, rules)
   action = rule.action
   return action
}
```

Model-based Reflex Agent

- Model describes how world evolves and effects of actions
- Where do model and rules come from?
- How to represent state and model?

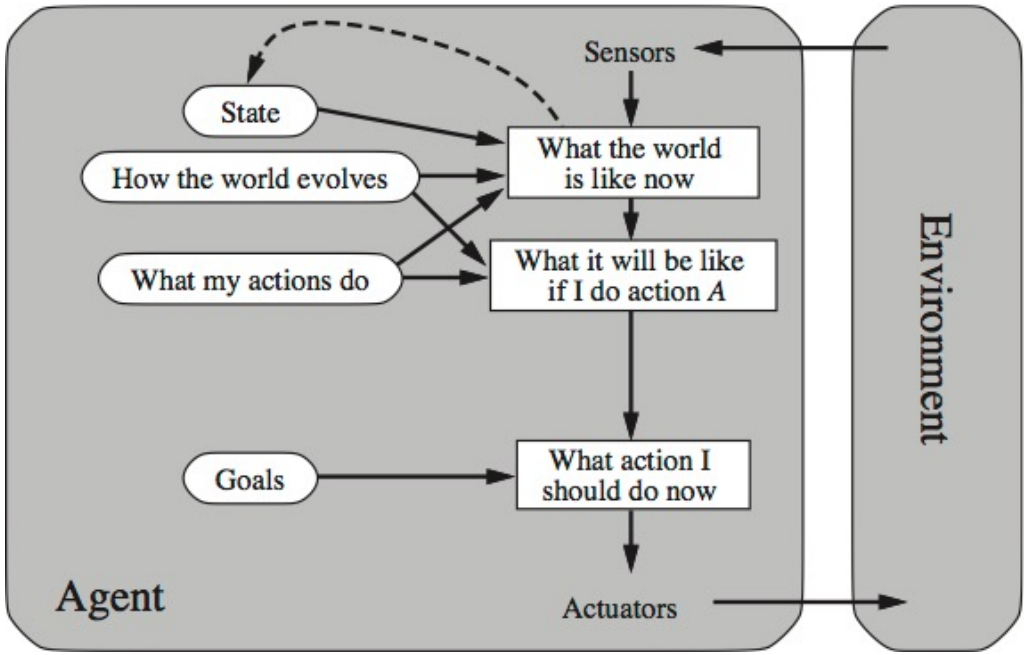



```
Action ModelBasedReflexAgent (Percept percept)
{
   RuleSet rules
   Model model

   state = UpdateState (state, action, percept, model)
   rule = RuleMatch (state, rules)
   action = rule.action
   return action
}
```

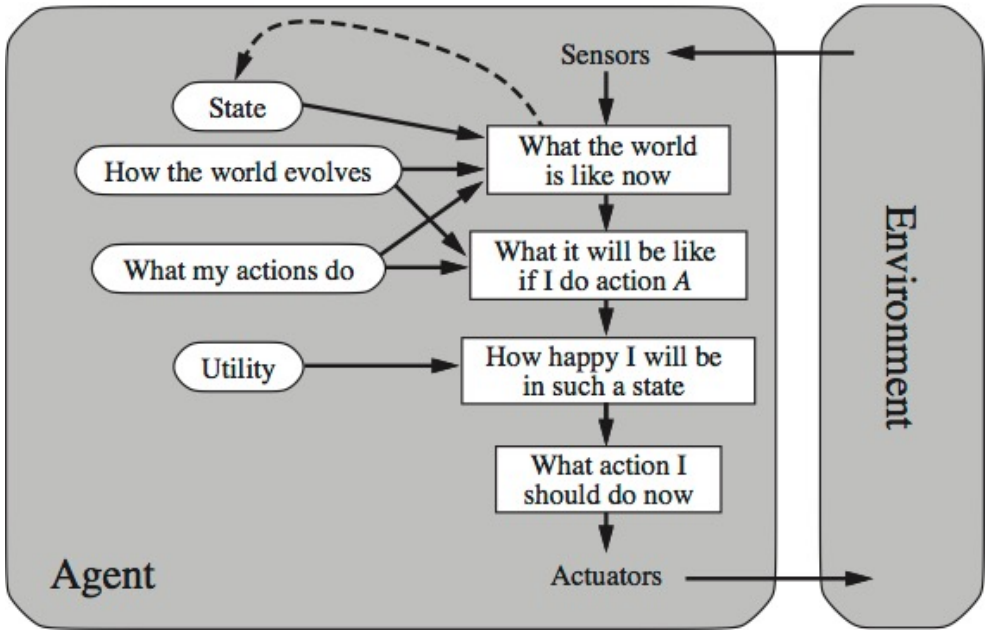
Goal-based Agent

- Search for sequence of actions to achieve goals
- Model, state, goals
  - Source?
  - Representation?



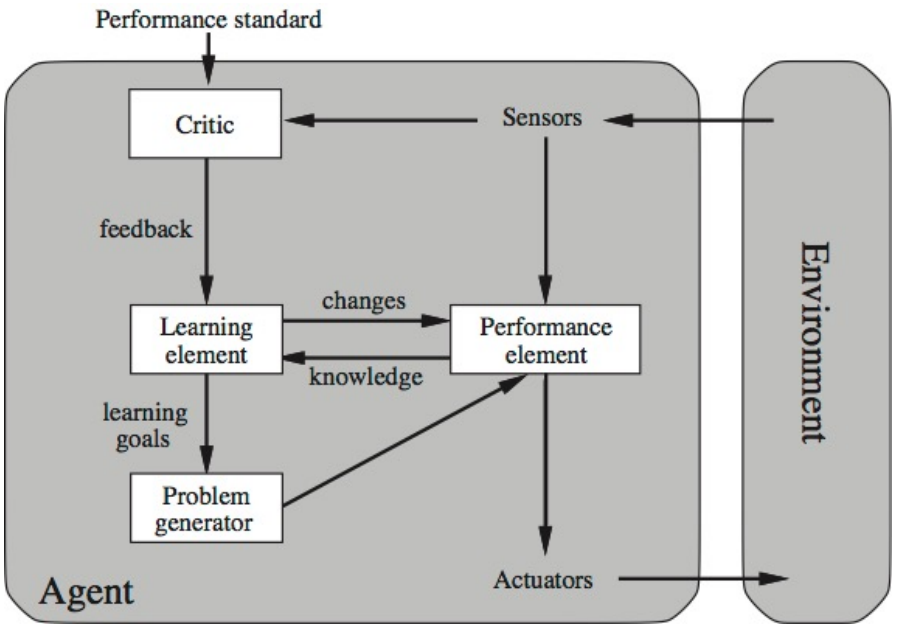
Utility-based Agent

- Search for sequence of actions to reach a high utility state
- Maximize expected utility
- Model, state, utility
  - Source?
  - Representation?



Learning Agent

- Learning element changes agent to improve performance
  - Models, rules, goals
- Performance element one of previous agents
- Critic provides feedback on agent's performance
- Problem generator drives agent to explore



Summary

- Rational agent seeks to maximize performance
- Agent's task defined in terms of **P**erformance, **E**nvironment, **A**ctuators and **S**ensors
- Agent's environment defined in terms of multiple dimensions (observability, …)
- Agent's type defined in terms of reflexes, rules, models, goals and/or utilities
- All agents can benefit from learning