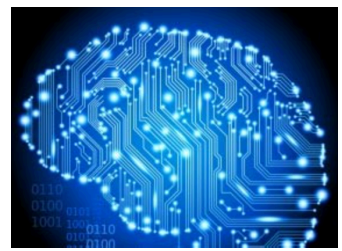
Lecture 1: Introduction

http://lamda.nju.edu.cn/yuy/course_ai16.ashx



What is artificial intelligence?

1956 Dartmouth meeting: "Artificial Intelligence"

John McCarthy:

" It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable."

1927-2011

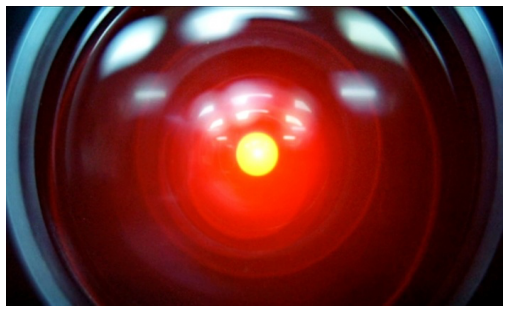
Marvin Minsky:

" to make computers be capable of doing things that when done by a human, would be thought to require intelligence "

1927-2016

we will discuss the concept and the history of AI in the last class

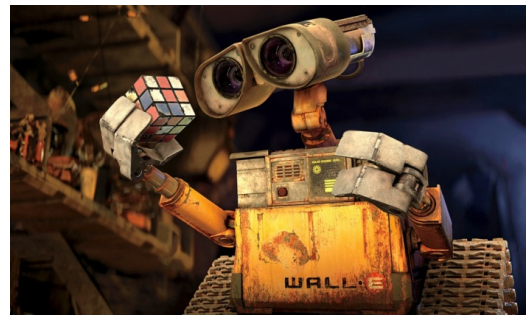
What we call AI in movies



2001: A Space Odyssey
1968

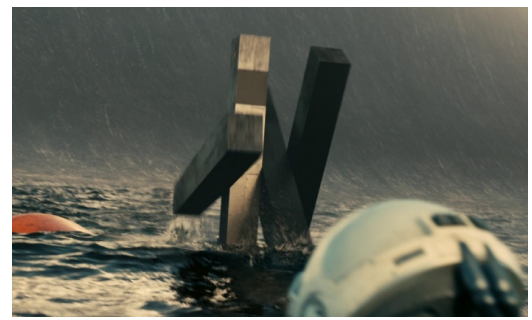
The Matrix
1999

A.I. Artificial Intelligence
2001



Wall-E
2008

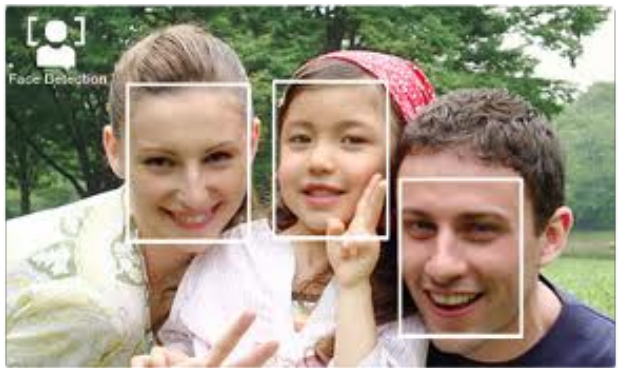
I, Robot
2004



Interstellar
2014

The Terminator
1984

What AI we do have



人脸检测、识别

S.I.R.I.

自动驾驶

推荐系统

下棋

BigDog

Current top AI systems

AlphaGo

AlphaGo v.s. 韩国职业选手李世乭（九段）

将于3月9、10、12、13、15日在Youtube上现场直播

Current top AI systems

Atlas

What we will learn

Search 搜索与规划

Knowledge 知识表达与处理

Uncertainty 不确定建模

Learning 机器学习

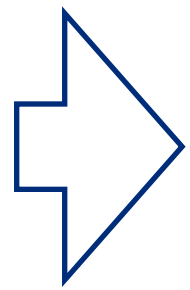
What we will do

Search 搜索与规划

Knowledge 知识表达与处理

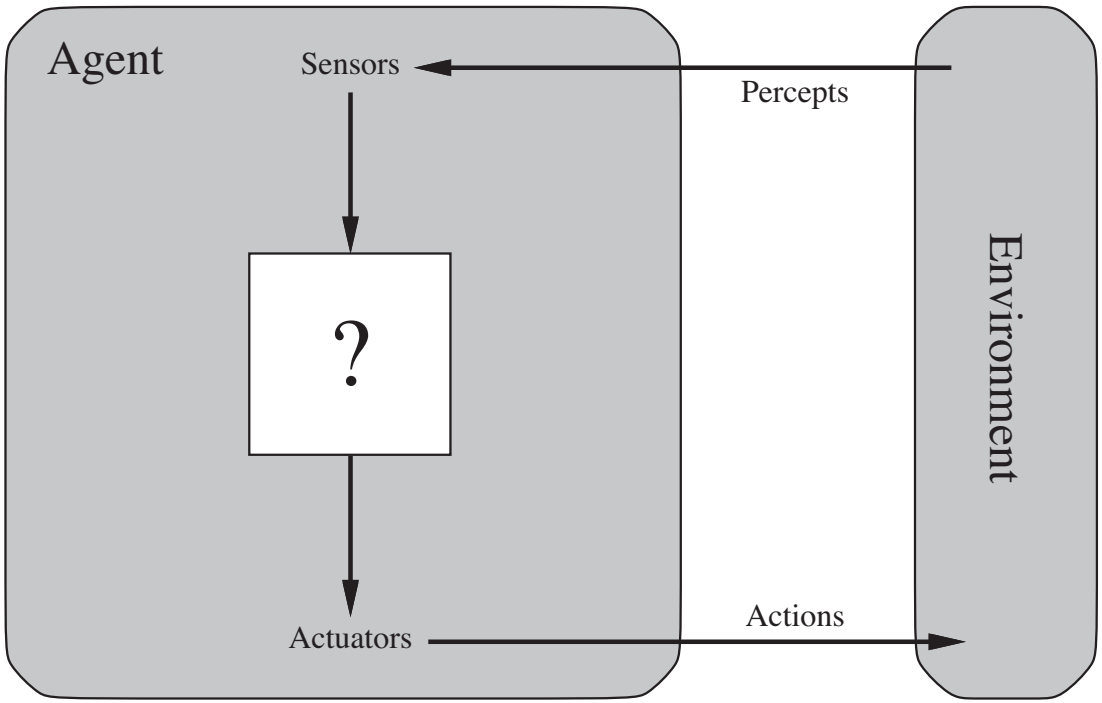
Uncertainty 不确定建模

Learning 机器学习



General
Game Player

Agent



Agents include humans, robots, softbots, thermostats, etc.

The agent function maps from percept histories to actions:

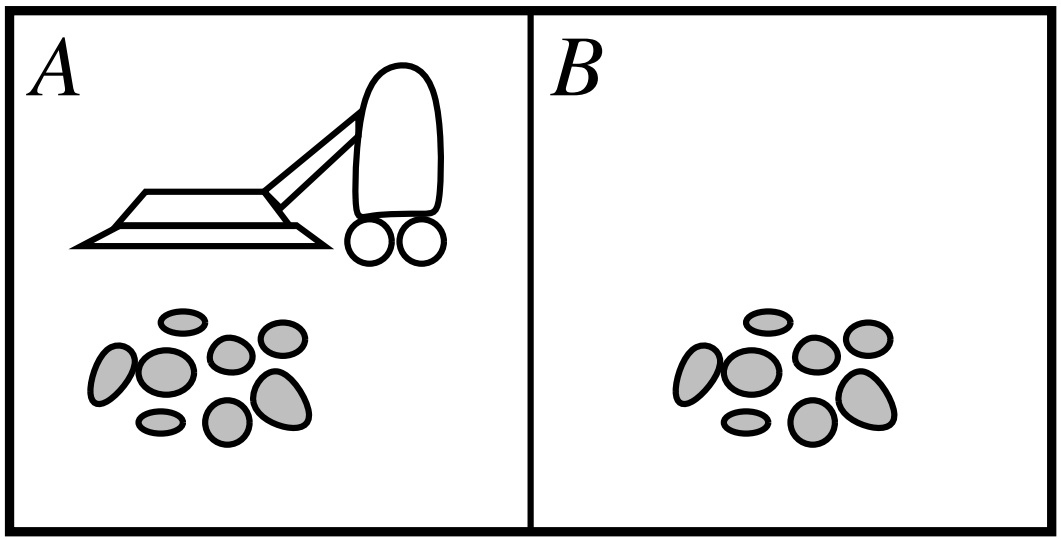
$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

The agent program runs on the physical architecture to produce f

Example: Vacuum-cleaner world

Percepts: location and contents, e.g., $[A, Dirty]$

Actions: $Left$, $Right$, $Suck$, $NoOp$



A vacuum-cleaner agent

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Dirty]$	$Suck$
⋮	⋮

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** $Left$

What is the **right** function?
Can it be implemented in a small agent program?

P. E. A. S.

To design an agent, we need to specify **four-dimensions**:

- Performance measure?
- Environment?
- Actuators?
- Sensors?

Examples of PEAS

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Environment types

In six-dimensions:

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
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

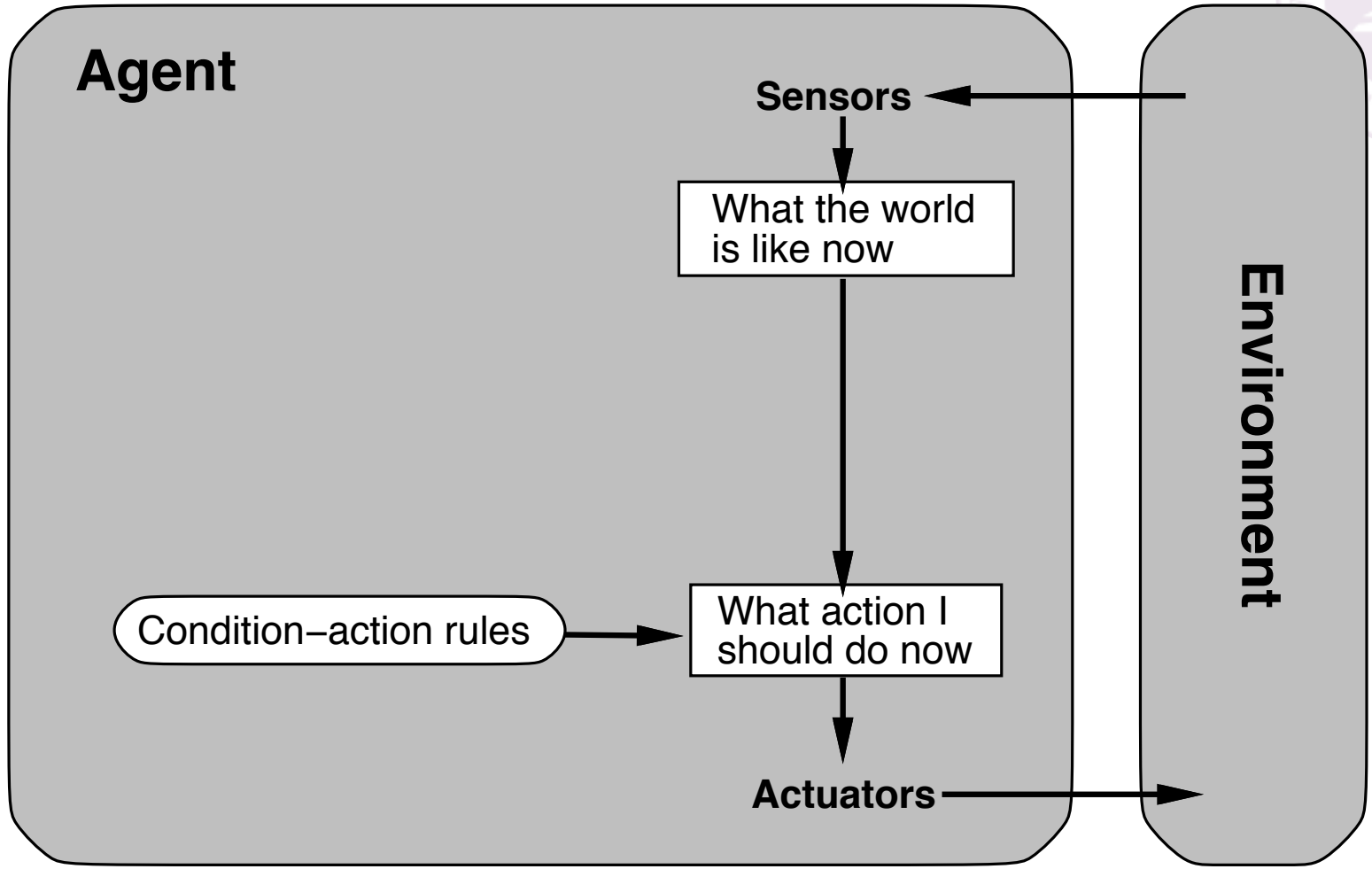
Agent types

Four basic types in order of increasing generality:

- simple reflex agents
- reflex agents with state
- goal-based agents
- utility-based agents

All these can be turned into learning agents

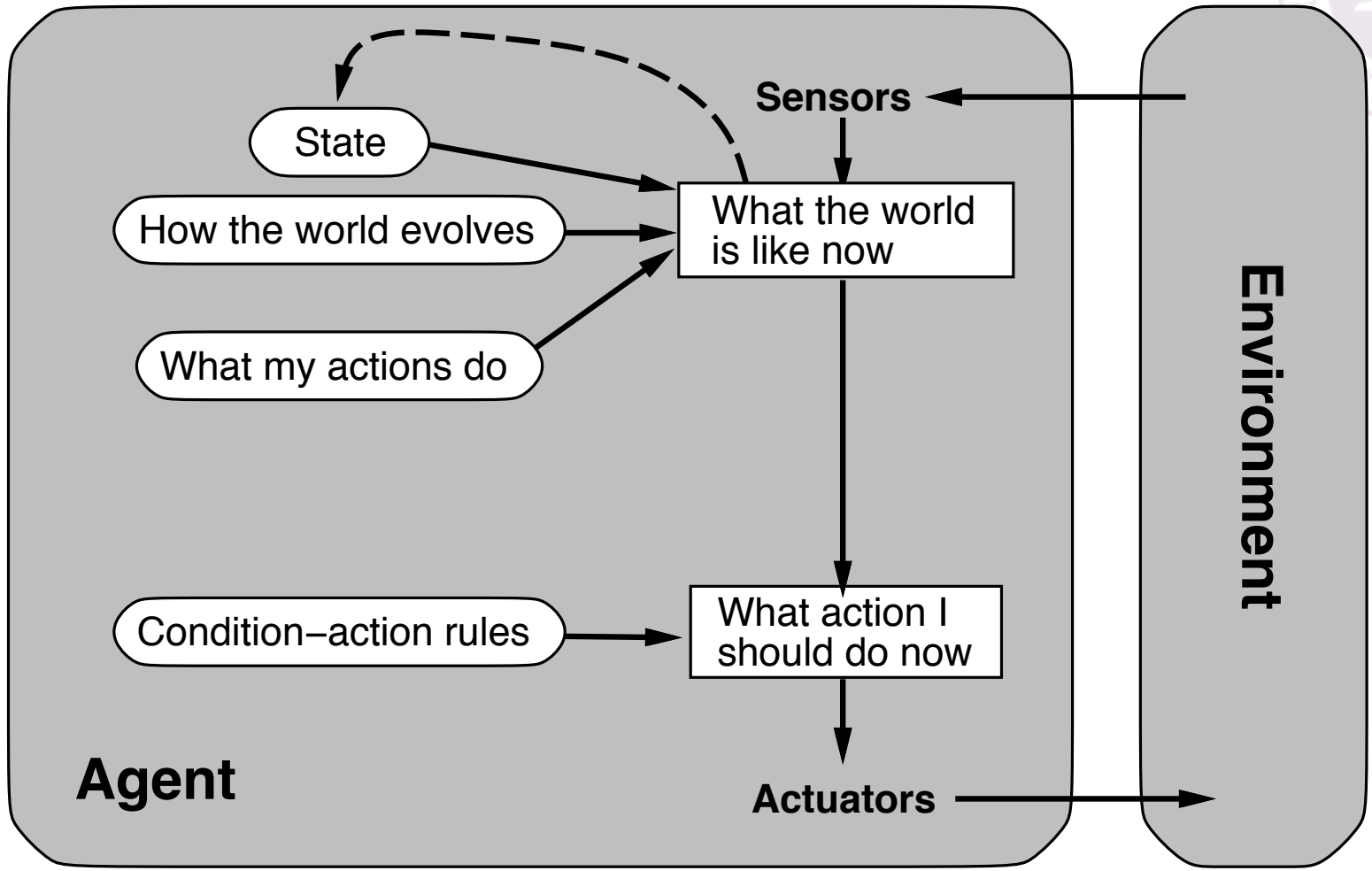
Simple reflex agents



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*

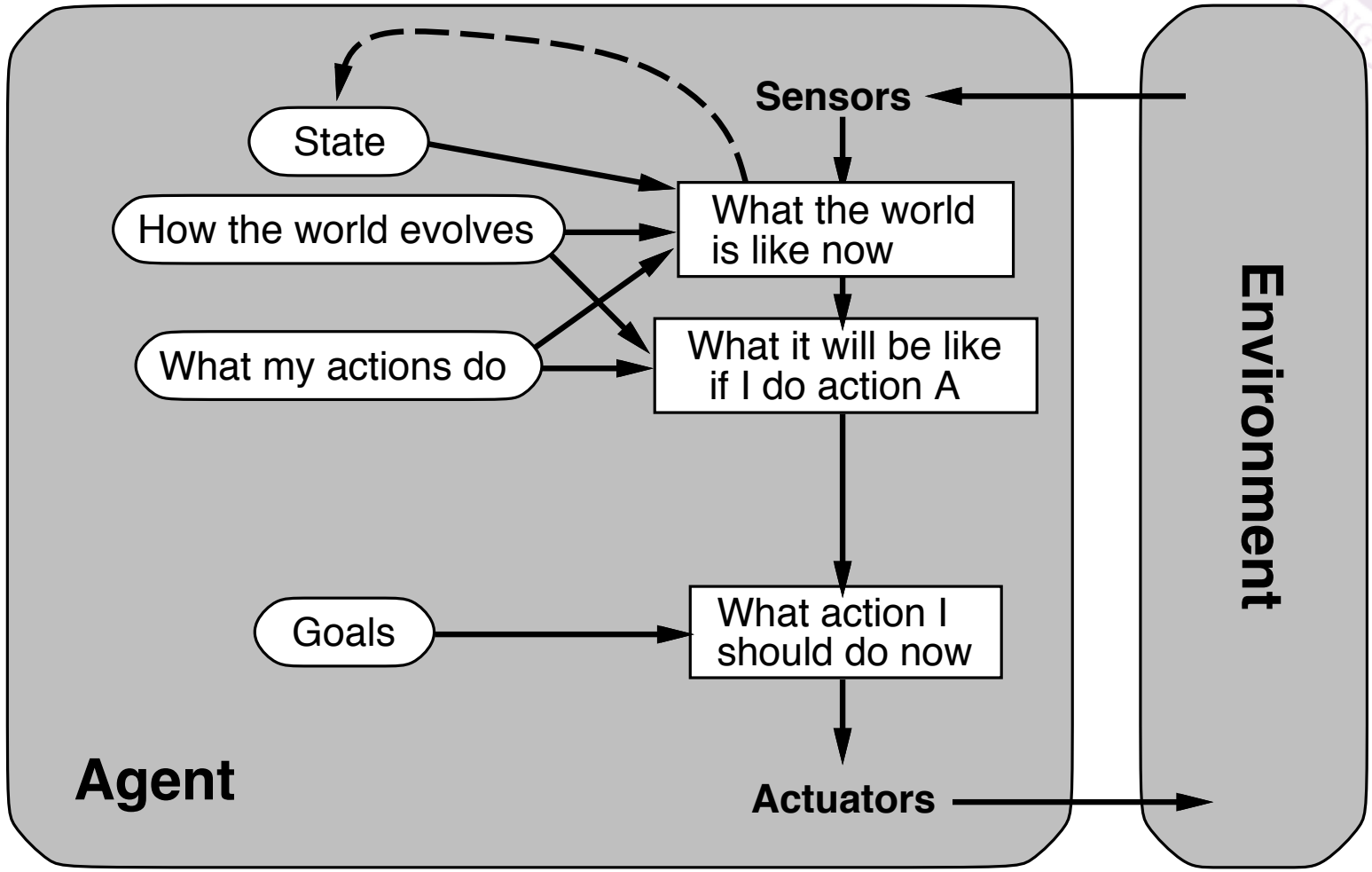
Reflex agents with state



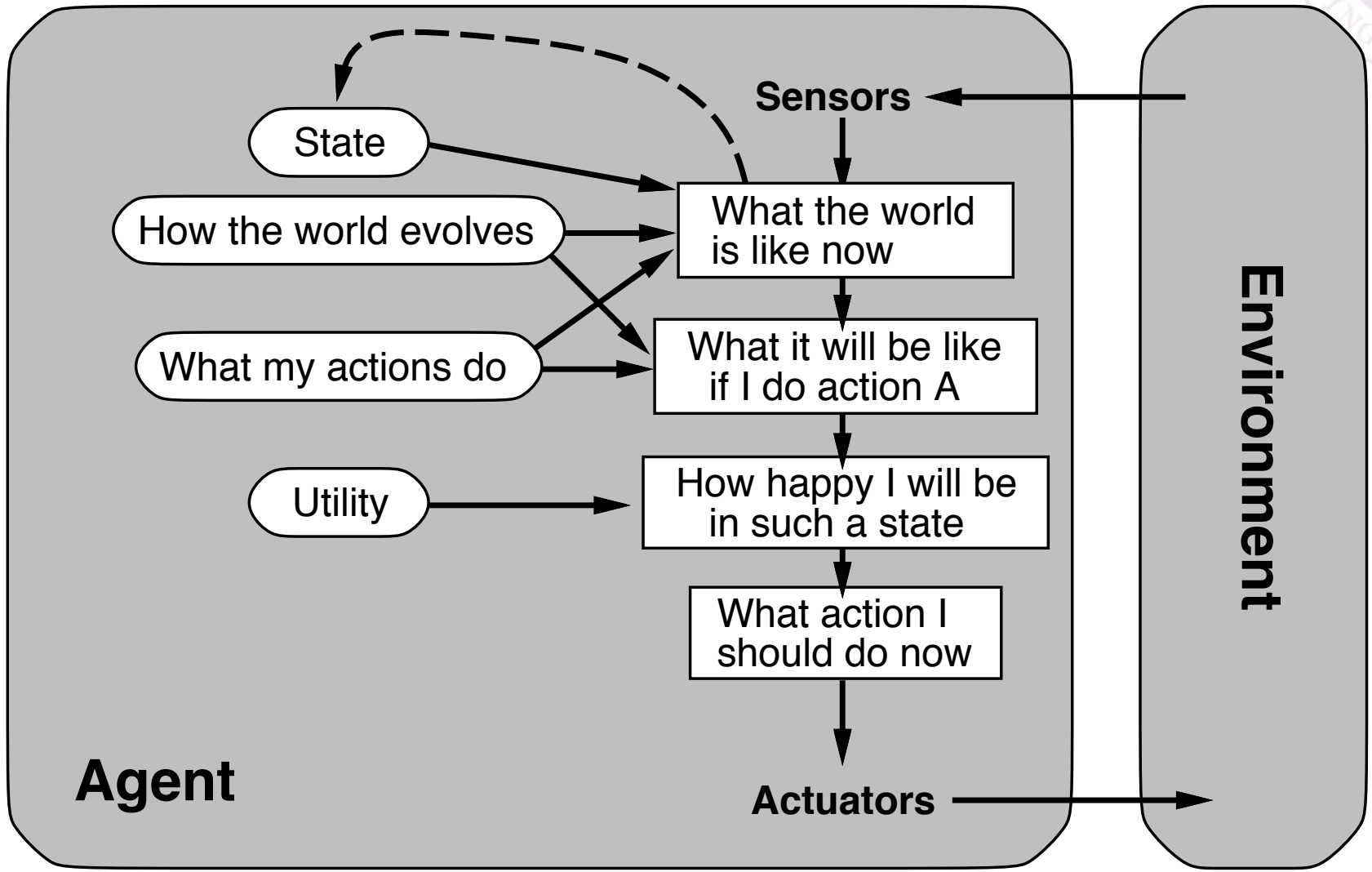
function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action
static: *last_A*, *last_B*, numbers, initially ∞

if *status* = *Dirty* **then** . . .

Goal-based agents



Utility-based agents



Learning agents