CIS471/571, Fall 2018
Introduction to Artificial Intelligence

Instructor: Dejing Dou

Office hours: Mondays 4:00pm-5:00pm

Sept. 26 and 28, 2018

GE-T: Gong (Eric) Zhang

The main topics

- This is an introduction course for Artificial Intelligence. The main topics we will try to cover:
  - what is AI?
  - intelligent agents
  - problem solving, search, game playing, CSP
  - logical agents, first order logic, knowledge representation
  - Uncertainty, probability, Bayes
  - machine learning (decision tree, ILP, Neural Network, Deep Learning)
  - NLP, information retrieval, information extraction
  - robotics

Evaluation

- Homework: 40%
  - There will be 5 or 6 assignments.
- Tests:
  - Midterm (Wednesday, 4-5:30pm Oct 31) 20%
  - Final (Tuesday, 2:45pm-4:45pm Dec 4) 20%
- Project: 20%
  - Part one (due Fri, Nov 16): A proposal for literature survey (CIS 471) or "AI" program design/implementation (CIS 571).
  - Part two (due Fri, Dec 7): A final paper with survey (CIS 471) or implementation/experiment report (CIS 571).

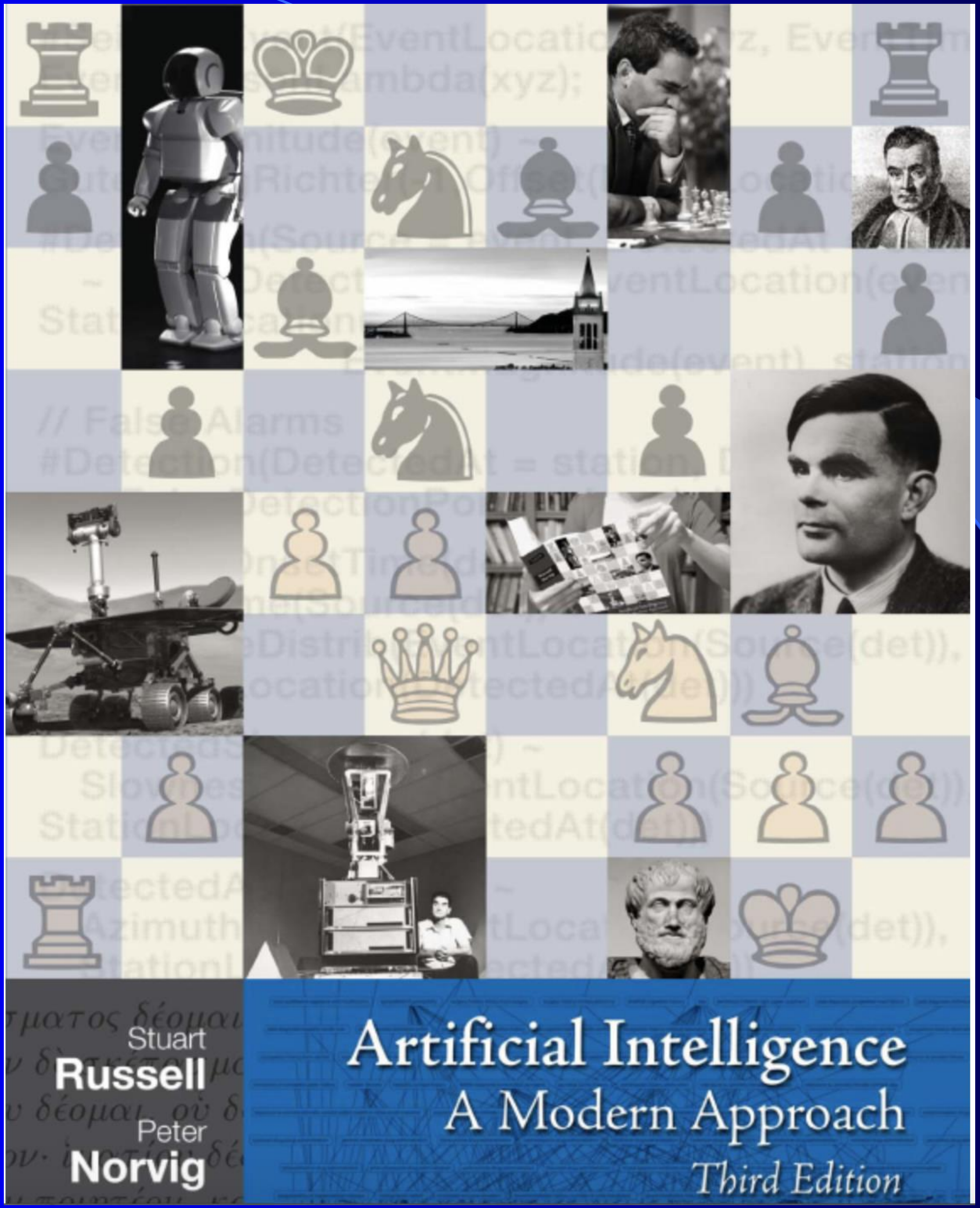
Textbook and course page

Stuart Russel and Peter Norvig, *Artificial Intelligence: A Modern Approach*, *3rd Edition*, Prentice Hall, 2010. (Available in UO bookstore)

Course Web Page:

http://www.cs.uoregon.edu/classes/18F/cis471/

Textbook


Stuart
Russell
Peter
Norvig
Artificial Intelligence
A Modern Approach
Third Edition

What is AI?

What is AI?

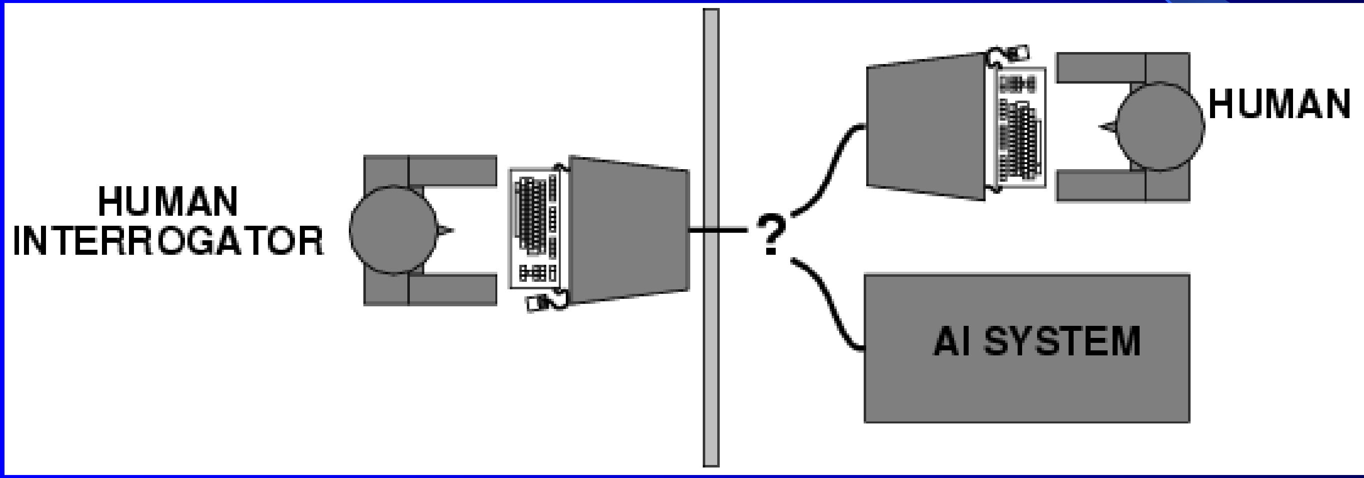
Views of AI fall into four categories:

Thinking humanly	Thinking rationally
Acting humanly	Acting rationally

- Thinking humanly: The cognitive modeling approach
- Acting humanly: The Turing Test
- Thinking rationally: Logic and Reasoning
- Acting rationally: The retional agent (e.g., a robot)
- The textbook advocates "acting rationally", however, *"thinking rationally" is more like what computer itself can do. "Acting humanly" (total Turing test) is more like what ordinary people expect from AI.*

Acting humanly: Turing Test

- Turing (1950) "Computing machinery and intelligence":
- "Can machines think?" → "Can machines behave intelligently?"
- Operational test for intelligent behavior: the Imitation Game



- Predicted that by 2000, a machine might have a 30% chance of fooling a lay person for 5 minutes (June 7, 2014, Russian chatterbot named Eugene Goostman, convinced 1/3 of judges that it was human)
- Anticipated all major arguments against AI in following 50 years
- Suggested major components of AI: knowledge, reasoning, language understanding, learning

Thinking humanly: cognitive modeling

- 1960s "cognitive revolution": information-processing psychology

- Requires scientific theories of internal activities of the brain. How human brain works.
- -- How to validate? Requires

  1) Predicting and testing behavior of human subjects (top-down)

  or 2) Direct identification from neurological data (bottom-up)

- Both approaches (roughly, Cognitive Science and Cognitive Neuroscience) are now distinct from AI.

Thinking rationally: "laws of thought"

- Aristotle: what are **correct** arguments/thought processes?
- Several Greek schools developed various forms of *logic*: *notation* and *rules of derivation* for thoughts;
- Direct line through mathematics and philosophy to modern AI --- *what computer itself can do (e.g., Logic Gates in circuit)*.
- Problems:

1. Not all intelligent behavior is mediated by logical inference, particularly when the knowledge is less than 100% certain.
2. What is the purpose of thinking? What thoughts should I have?
3. *Is thinking = reasoning?*

Acting rationally: rational agent

- Rational behavior: doing the **right** thing

- The right thing: that which is expected to *maximize* goal achievement, given the available information
- Doesn't necessarily involve thinking – e.g., blinking reflex – but thinking should be in the service of rational action.
- The textbook advocates "acting rationally."

Rational agents

- An agent is an entity that perceives and acts.
  - Hardware agent: Robots
  - Software agent: Web agents and Chatbots
- This textbook is about designing rational agents
- Abstractly, an agent is a function from percept histories to actions:
-

$$[f: \mathcal{P}^{\star} \rightarrow \mathcal{A}]$$

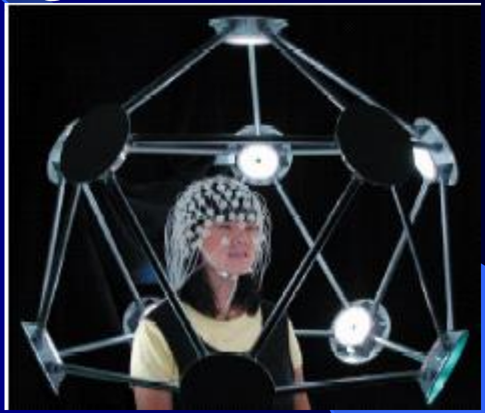
- For any given class of environments and tasks, we seek the agent (or class of agents) with the best performance
-
- Note: computational limitations make perfect rationality unachievable → design best program for given machine resources
-

The Foundations of AI

Philosophy	Logic, methods of reasoning, mind as physical system foundations of learning, language, rationality
Mathematics	Formal representation and proof algorithms, computation, (un)decidability, (in)tractability, probability (uncertainty)
Economics	utility, decision theory, game theory
Neuroscience	physical substrate for mental activity
Psychology	phenomena of perception and motor control, experimental techniques
Computer engineering	building fast computers
Control theory	design systems that maximize an objective function over time
Linguistics	knowledge representation, grammar

Some interesting examples

- Mathematics: Cook and Karp showed the existence of large classes of canonical combinatorial search and reasoning problems that are NP-complete.
- Neuroscience: EEG and fMRI are giving images of brain activity (e.g., NIC and EGI @ UO).



- Control theory: before self-control machine (e.g., a water clock with the regulator), only living things can modify their behavior in response to changes in environment.
- Linguistics: Syntax translation and Semantic translation between two languages (e.g., google translation).

History of AI

- 1943 McCulloch & Pitts: Boolean circuit model of brain
- 1950 Turing's "Computing Machinery and Intelligence"
- 1956 Dartmouth meeting: "Artificial Intelligence" adopted,
- 1952—69 Look, Ma, no hands!
- 1950s Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
- 1965 Robinson's complete algorithm for logical reasoning
- 1966—73 AI discovers computational complexity
  Neural network research almost disappeared, then
- 1969—79 Early development of KB systems, Expert systems
- 1980-- AI becomes an industry
- 1986-- Neural networks return to popularity, then cool again
- 1987-- AI becomes a science
- 1995-- The emergence of intelligent agents
- 2000-- AI and Big Data (Big Data, IoT)
- 2010-- Deep Learning

Milestones of AI applications

- During the 1991 Gulf War, US forces deployed an AI logistics planning and scheduling program that involved up to 50,000 vehicles, cargo, and people
- Proved a mathematical conjecture (Robbins conjecture) unsolved for decades 1996
- Deep Blue defeated the reigning world chess champion Garry Kasparov in 1997
- No hands across America (driving autonomously 98% of the time from Pittsburgh to San Diego) in 2003
- NASA's on-board autonomous planning program controlled the scheduling of operations for a spacecraft and Mars Exploration Rovers 2008
- Since 2010, autonomous cars (e.g., Google Car) become legal for self-driving in several states in US.
- In 2011, the Watson computer system competed on *Jeopardy!* against former winners Brad Rutter and Ken Jennings winning the first place prize of $1 million

- AlphaGo beat former GO world champion Lee Sedol 4:1 in 2016

State of Arts

- Watson Project (QA): Developed by IBM DeepQA, In 2011, Watson competed on *Jeopardy!* against former winners Brad Rutter and Ken Jennings. Watson received the first prize of $1M.
- SIRI (chatbot): Apple App 2010 (Robot take over the world) ➔ Matrix
- Google Car: Legally driving in CA, 2012
- Atlas: Boston Dynamics Humanoid Robots
- Deep Learning: Since 2006, Return of multi-layer ANN. With large amount training data（Big Data）and computational resources, deep learning systems achieves close to 100% accuracy in face recognition, speech recognition, and handwriting recognition tasks.
- AlphaGo, developed by DeepMind, beat former GO world champion Lee Sedol 4:1 in 2016

IBM Watson (QA)

Deep Blue (Chess)

AlphaGo (GO)

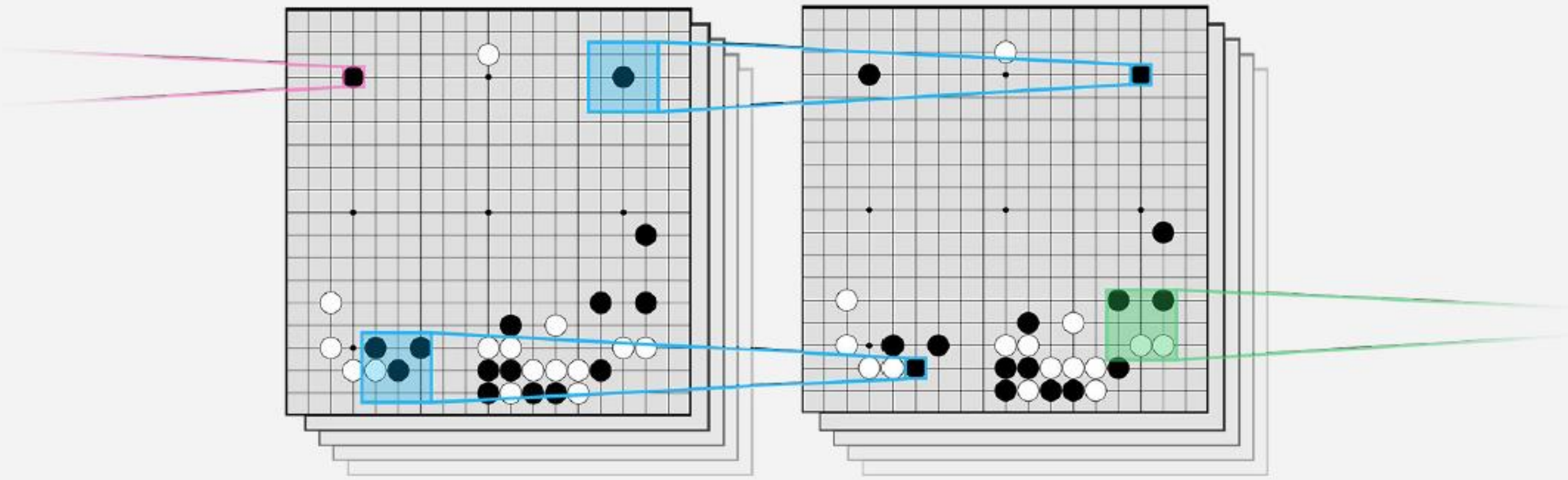
Why is Go hard for computers to play?

Game tree complexity = b^d

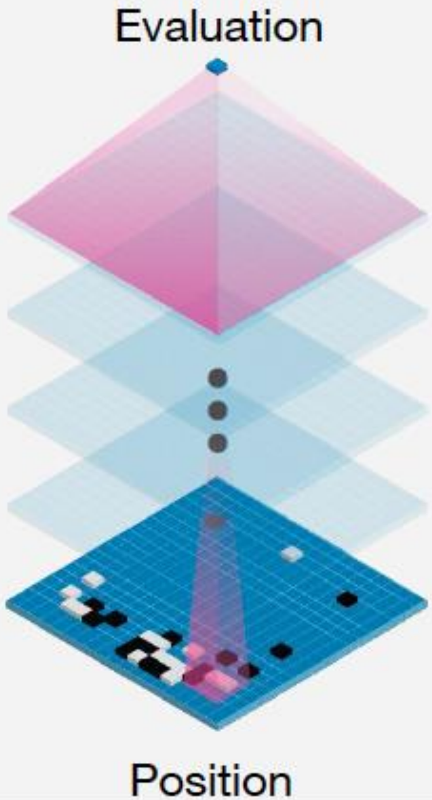
Brute force search intractable:

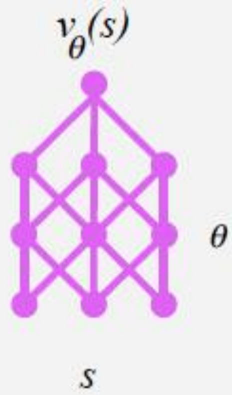
1. Search space is huge
2. “Impossible” for computers to evaluate who is winning

Convolutional neural network

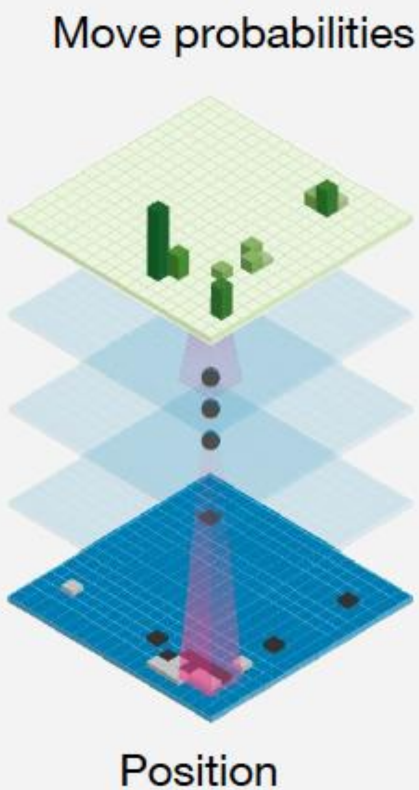


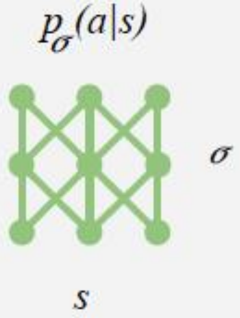
Value network



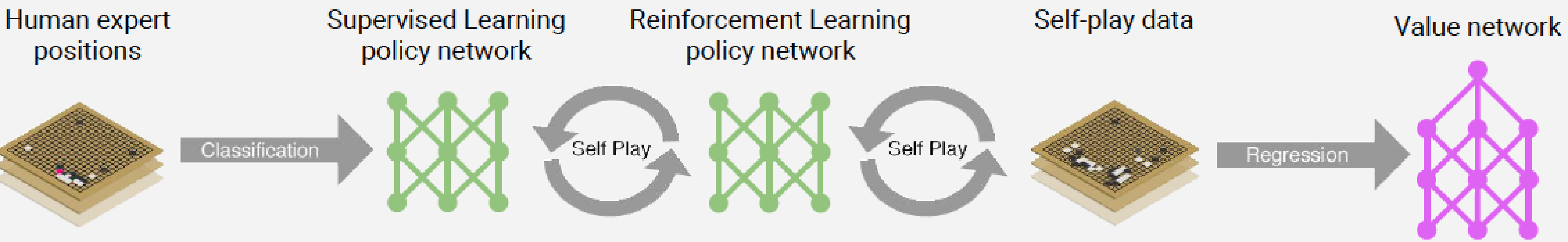


Policy network





Neural network training pipeline



Nature Cover

AlphaGo Zero

AlphaGo Zero

ARTICLE

doi:10.1038/nature24270

Mastering the game of Go without human knowledge

David Silver[1]*, Julian Schrittwieser[1]*, Karen Simonyan[1]*, Ioannis Antonoglou[1], Aja Huang[1], Arthur Guez[1], Thomas Hubert[1], Lucas Baker[1], Matthew Lai[1], Adrian Bolton[1], Yutian Chen[1], Timothy Lillicrap[1], Fan Hui[1], Laurent Sifre[1], George van den Driessche[1], Thore Graepel[1] & Demis Hassabis[1]

A long-standing goal of artificial intelligence is an algorithm that learns, *tabula rasa*, superhuman proficiency in challenging domains. Recently, AlphaGo became the first program to defeat a world champion in the game of Go. The tree search in AlphaGo evaluated positions and selected moves using deep neural networks. These neural networks were trained by supervised learning from human expert moves, and by reinforcement learning from self-play. Here we introduce an algorithm based solely on reinforcement learning, without human data, guidance or domain knowledge beyond game rules. AlphaGo becomes its own teacher: a neural network is trained to predict AlphaGo's own move selections and also the winner of AlphaGo's games. This neural network improves the strength of the tree search, resulting in higher quality move selection and stronger self-play in the next iteration. Starting *tabula rasa*, our new program AlphaGo Zero achieved superhuman performance, winning 100–0 against the previously published, champion-defeating AlphaGo.

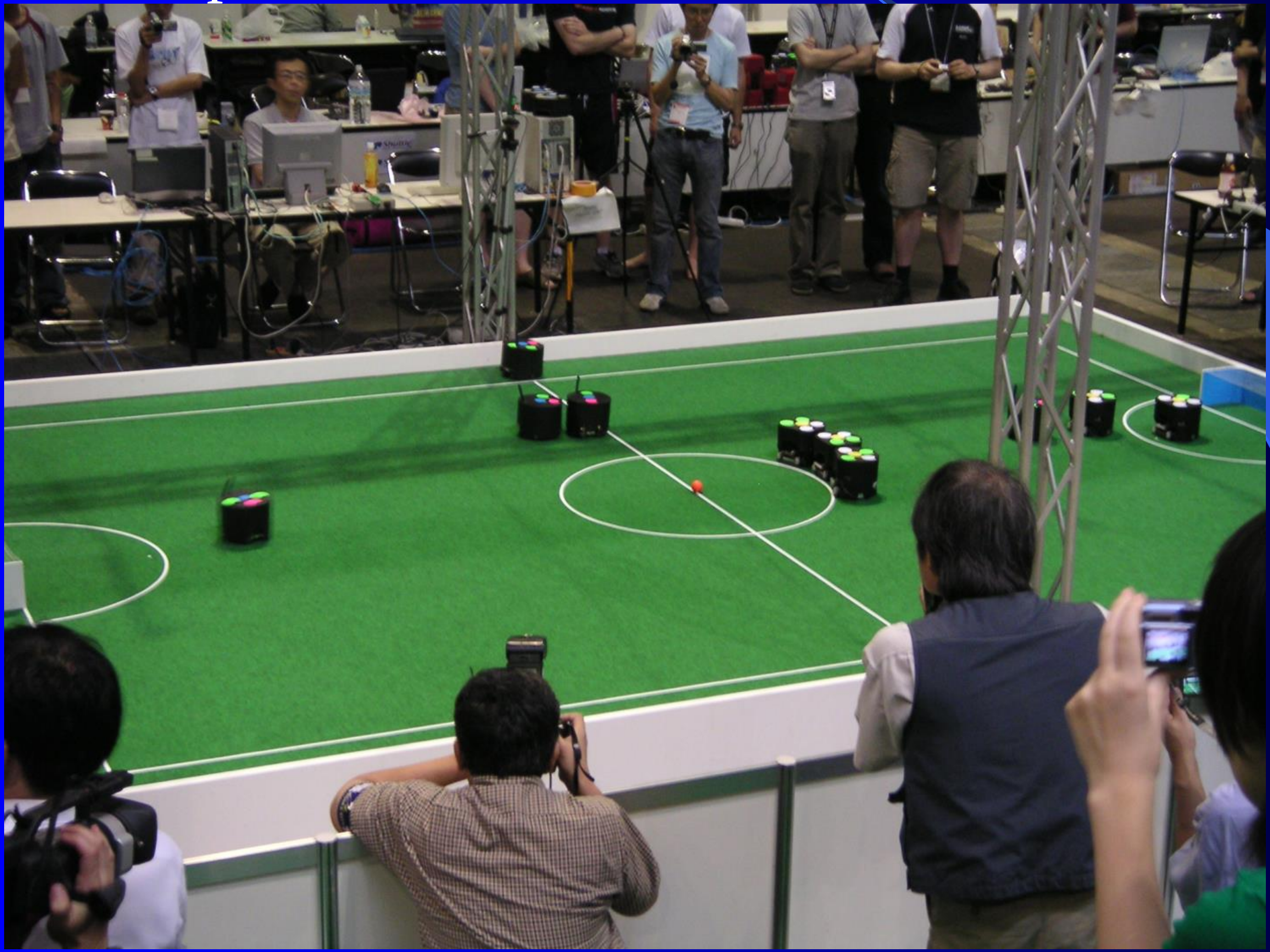
Google Car (Automatic Driving)

Football Version of Turing Test

- *By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world football/soccer champion team.*

State of the art (Robotics)

- *By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team.*



Sony Dogs

Humanoid Robots

Boston Dynamics Atlas

General AI

- What is General AI?
  - the intelligence of a machine that could successfully perform any intellectual task that a human being can. GAI is also referred to as "strong AI" or "full AI."
  - i.e., passing full Turing test

- How far are we from General AI?

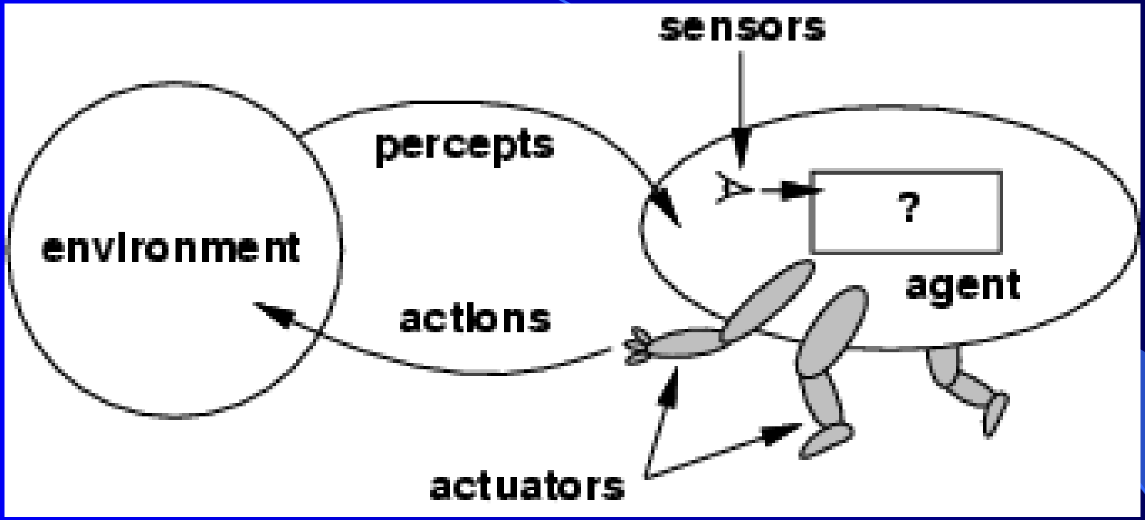
Chapter 2: Intelligent Agents

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

What is (intelligent) Agent?

- 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
- Software agent: receive keystrokes, files contents and network packets, and sending files contents and network packets.

Agents and environments

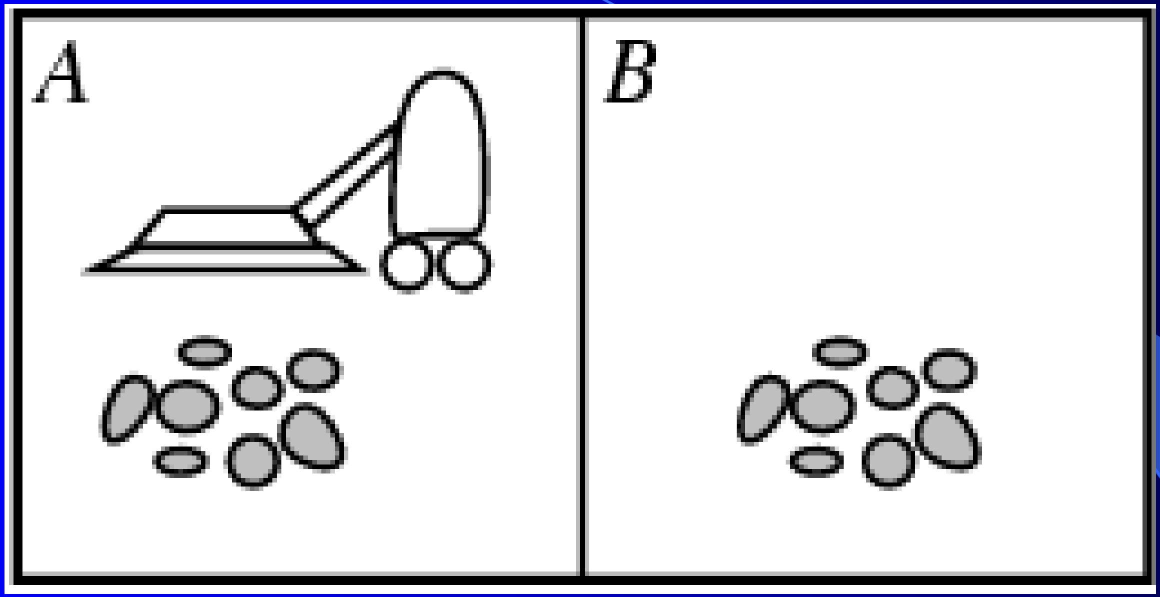


- The agent function maps from percept histories to actions:

$$[f: \mathcal{P}^\star \rightarrow \mathcal{A}]$$

- The agent program runs on the physical architecture to implement
- agent = architecture + program

Vacuum-cleaner world



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

A vacuum-cleaner agent

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

Rationality

- 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.
  - How about vacuum-cleaner?
- 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

- A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date.

Rational != omniscient

(percepts may not supply all relevant information)

Rational != perfection

(action outcomes may not be as expected, *perfection is try to maximize the expected performance, rationality maximize the actual performance*)

Hence, rational != successful

Rational = exploration, learning, autonomy (extent that relies on its own percepts)

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 (e.g., google car):
-
  - Performance measure: Safe, fast, legal, comfortable trip, maximize profits
  -
  - Environment: Roads, other traffic, pedestrians, customers
  -
  - Actuators: Steering wheel, accelerator, brake, signal, horn
  -
  - Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard

PEAS

- Agent: Medical diagnosis system (softbot)
- Performance measure: increase healthy patients, 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: Instructors (human agent) (e.g., *Dejing*)
- Performance measure: Maximize knowledge students get and student's score on exams
- Environment: Set of students, Classroom, Office
- Actuators: Lectures, Exercises, Office hours. *(Can "make exams easy" work?)*
- Sensors: Anyway can be used to judge performance of students (e.g., grading)

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.
- 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 (cont’d)

- Static (vs. dynamic): The environment is unchanged while an agent is deliberating.
- 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.
-

Agent functions and programs

- An agent is completely specified by the agent function mapping percept sequences to actions

$$[f: \mathcal{P}^\star \rightarrow \mathcal{A}]$$

- The agent program runs on the physical architecture to implement f
-
- One agent function (or a small equivalence class) should be rational
-
- Aim: find a way to implement the rational agent function concisely (by agent program)

Agent program for a vacuum-cleaner agent

function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action

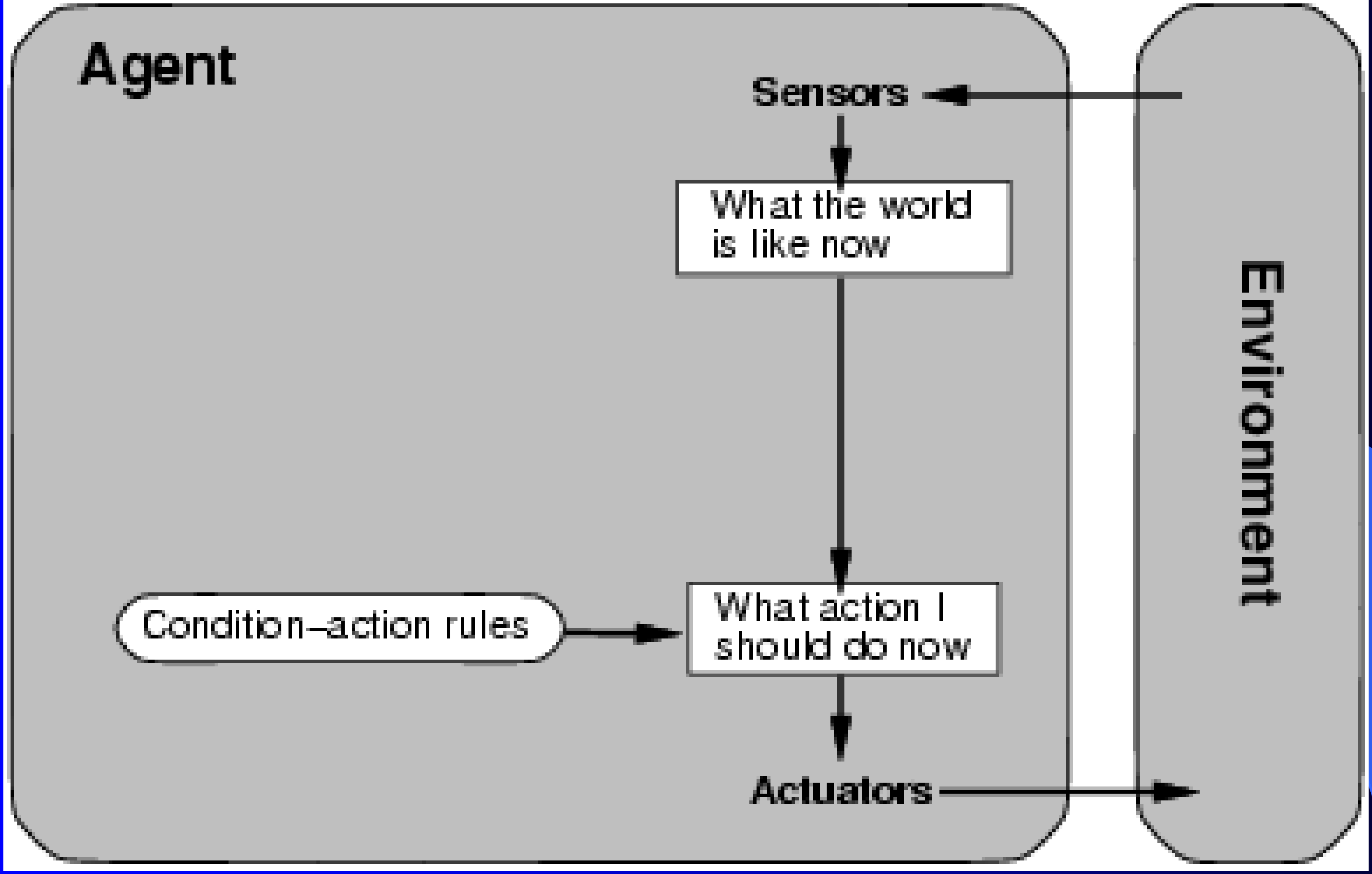
  if *status* = *Dirty* **then return** *Suck*
  else if *location* = *A* **then return** *Right*
  else if *location* = *B* **then return** *Left*

Agent types

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

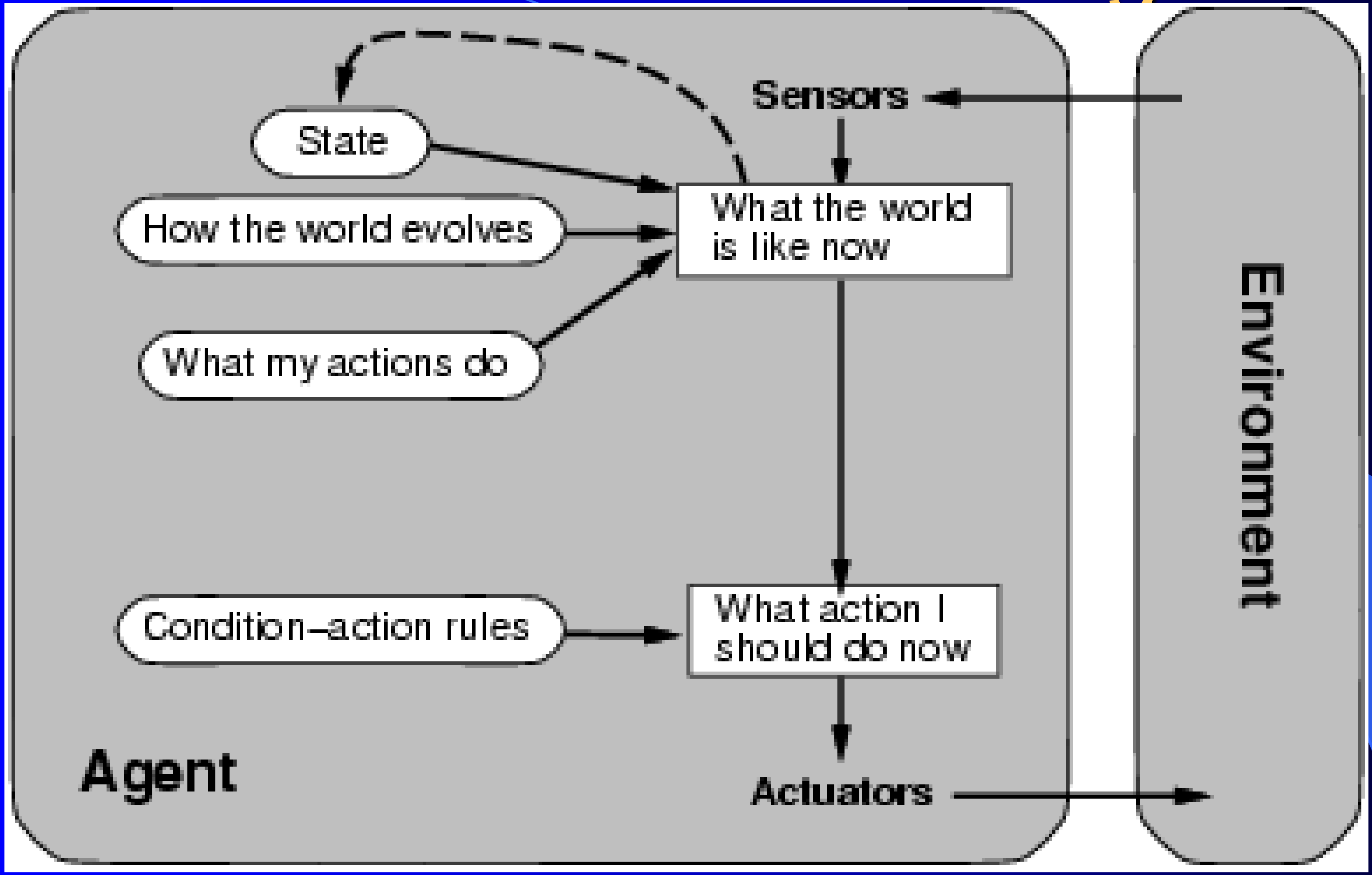
- Learning agents
-

Simple reflex agents



state <= Interpret-Input (*percept*)
rule <= Rule-Match (*state*, *rules*)
action <= Rule-Action [*rule*]

Model-based reflex agents

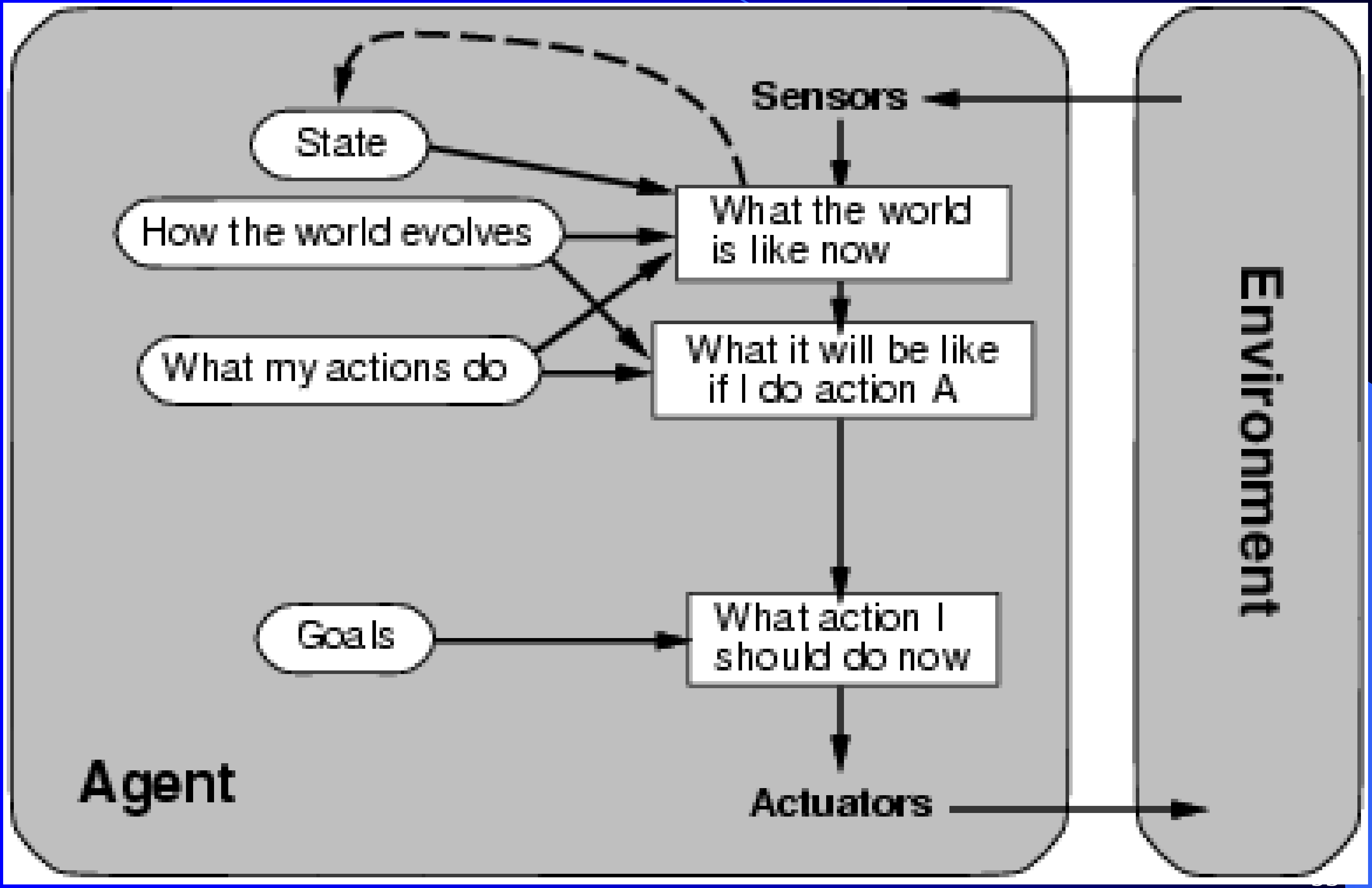


state <= Update-State (*state*, *action*, *percept*)

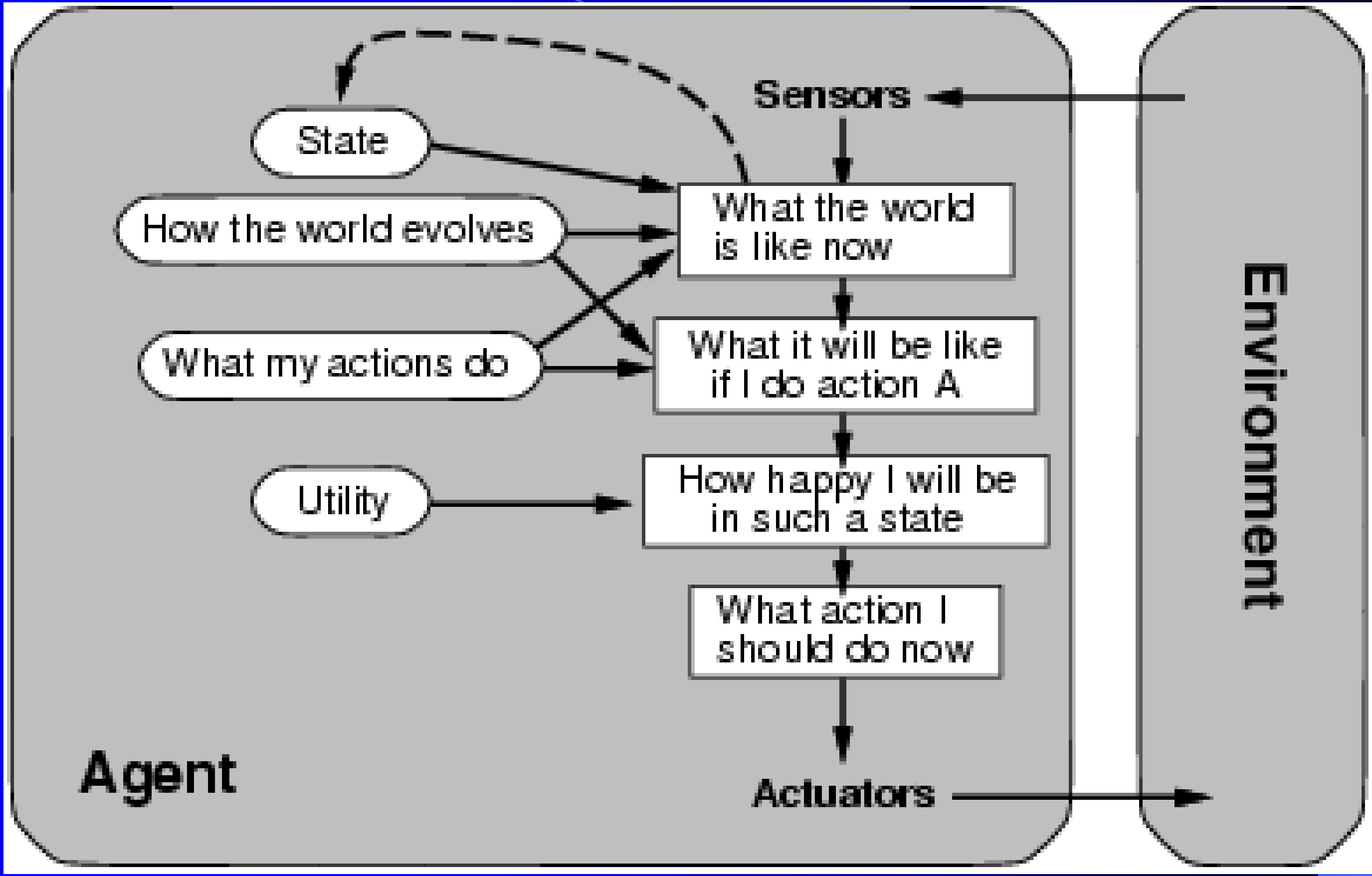
rule <= Rule-Match (*state*, *rules*)

action <= Rule-Action [*rule*]

Goal-based agents

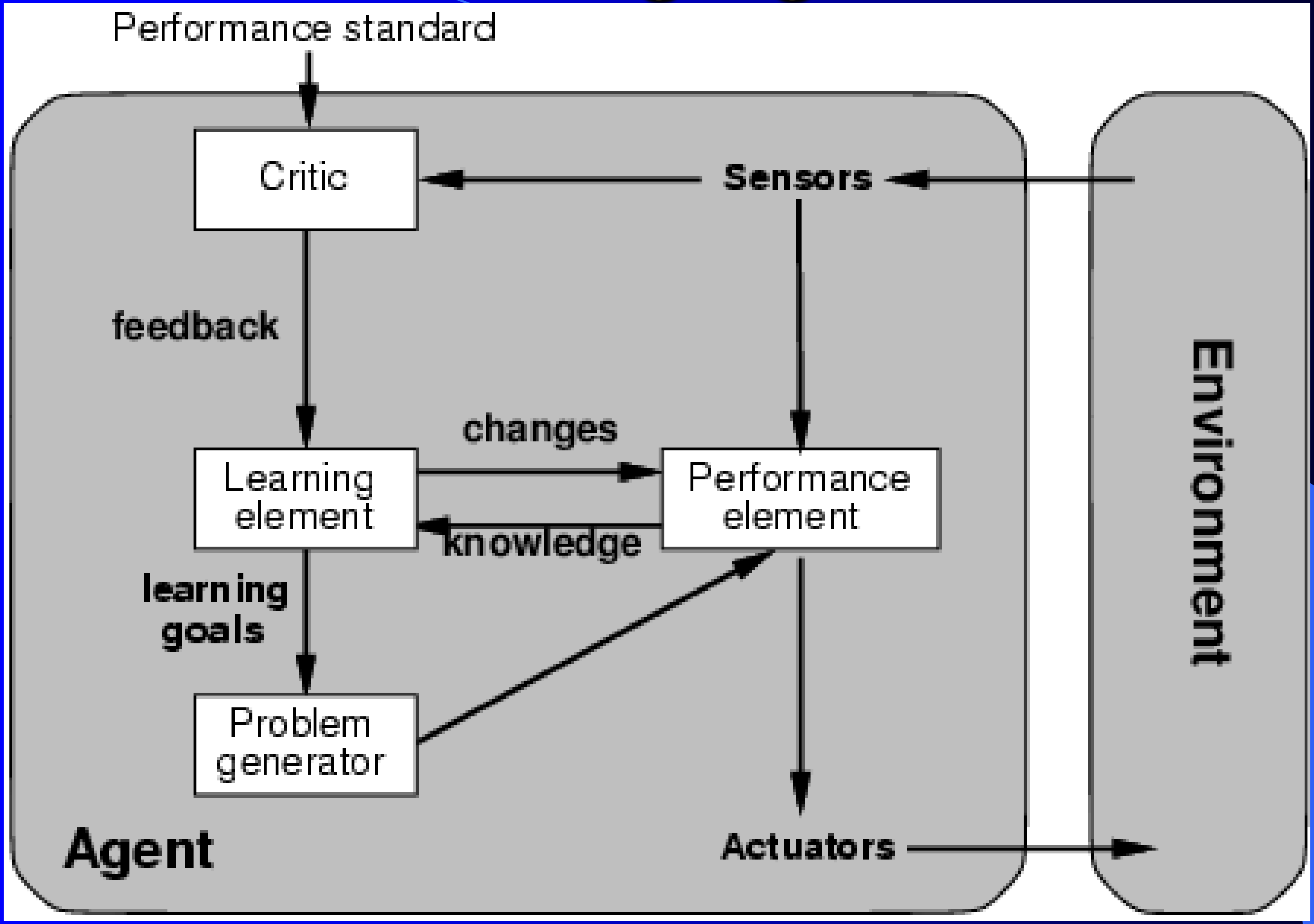


Utility-based agents



A utility function maps a state(s) to onto a real number

Learning agents



Note: Performance element selects actions and Problem generator suggests actions that will lead to new experience.