COGS500/CMPE489 Introduction to Cognitive Science Week II: Artificial Intelligence

Emre Uğur
Computer Engineering
Bogazici University

With slides adapted from Albert Ali Salah

Cem Say’s Lecture

- Artificial Intelligence in Dartmouth Workshop (1956)
  - vs. Cybernetics of Norbert Wiener
  - by John McCarthy, Marvin Minsky, Claude Shannon, Allen Newell, Herbert Simon (e.g. logic theorist)
- Alan Turing:
  - Entscheidungsproblem (Decision Problem), Formal Definition of Algorithm
  - Turing Machine
  - Universal Turing Machine
  - Turing Test
  - A. M. Turing (1950) Computing Machinery and Intelligence. Mind 49: 433-460.
  - Philosophical Discussions - Searle, Dennett

Quiz:
1- Which parts are finite and infinite in Turing Machines?
2- What is a Universal Turing Machine?

- Alan Turing:
  - Entscheidungsproblem (Decision Problem), Formal Definition of Algorithm
  - Turing Machine
  - Universal Turing Machine
  - Turing Test
  - A. M. Turing (1950) Computing Machinery and Intelligence. Mind 49: 433-460.
  - Philosophical Discussions - Searle, Dennett

Early enthusiasm (1952-69)

- Claims: computers can do X
- *General Problem Solver*, Newell & Simon
  - Intentionally solved puzzles in a similar way as humans do (order of subgoals, etc)
- *Geometry Theorem Prover*, Herbert Gelernter, 1959
- Arthur Samuel’s learning checkers program 1952
- LISP, time sharing, Advice taker: McCarthy 1958
- Integration, IQ geometry problems, algebra stories
- Blocks world: vision, learning, NLP, planning
- Adalines [Widrow & Hoff 1960], perceptron convergence theorem [Rosenblatt 1962]

AI Programming Languages

- LISP (by John McCarty in 1958)
  - influenced by the notation of lambda calculus.
  - List processing and recursion
  - natural language understanding and generation,
  - game playing,
  - theorem proving,
  - early computer vision,
  - symbolic mathematics ,
  - problem-solving and knowledge representation, were produced at Stanford and MIT.

AI Programming Languages

- Prolog (~1970)
  - First order logic: facts and rules

```
mother_child(trude, sally).

father_child(tom, sally).
father_child(tom, erica).
father_child(mike, tom).

sibling(X, Y)      :- parent_child(Z, X), parent_child(Z, Y).

parent_child(X, Y) :- father_child(X, Y).
parent_child(X, Y) :- mother_child(X, Y).
```

```
?- sibling(sally, erica).
Yes
```

AI Programming Languages

- LISP (by John McCarty in 1958)
  - influenced by the notation of lambda calculus.
  - List processing and recursion
- ELIZA (1966): Fortran-like language mimics lisp

ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine

JOSEPH WEIZENBAUM
Massachusetts Institute of Technology, Cambridge, Mass.*

https://www.masswerk.at/elizabot/eliza.html

Men are all alike.
IN WHAT WAY
They're always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE
Well, my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE
He says I'm depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED

AI Programming Languages

- LISP (by John McCarty in 1958)
  - influenced by the notation of lambda calculus.
  - List processing and recursion
- LISP-based SHRDLU (natural language understanding program) in 1970

Person: Pick up a big red block.

Computer: OK.

Person: Grasp the pyramid.

Computer: I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.

Person (changing their mind): Find a block which is taller than the one you are holding and put it into the box.

Computer: BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.

Computer: OK.

Person: What does the box contain?

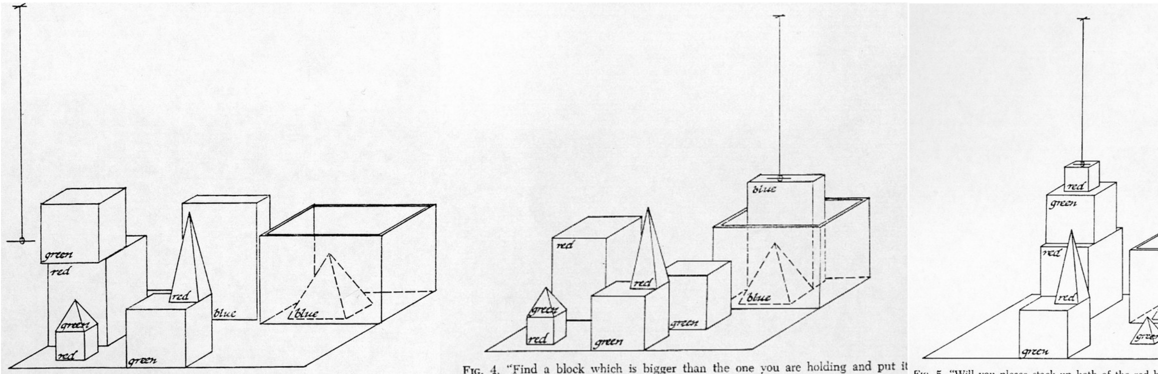
Computer: THE BLUE PYRAMID AND THE BLUE BLOCK.

Person: What is the pyramid supported by?

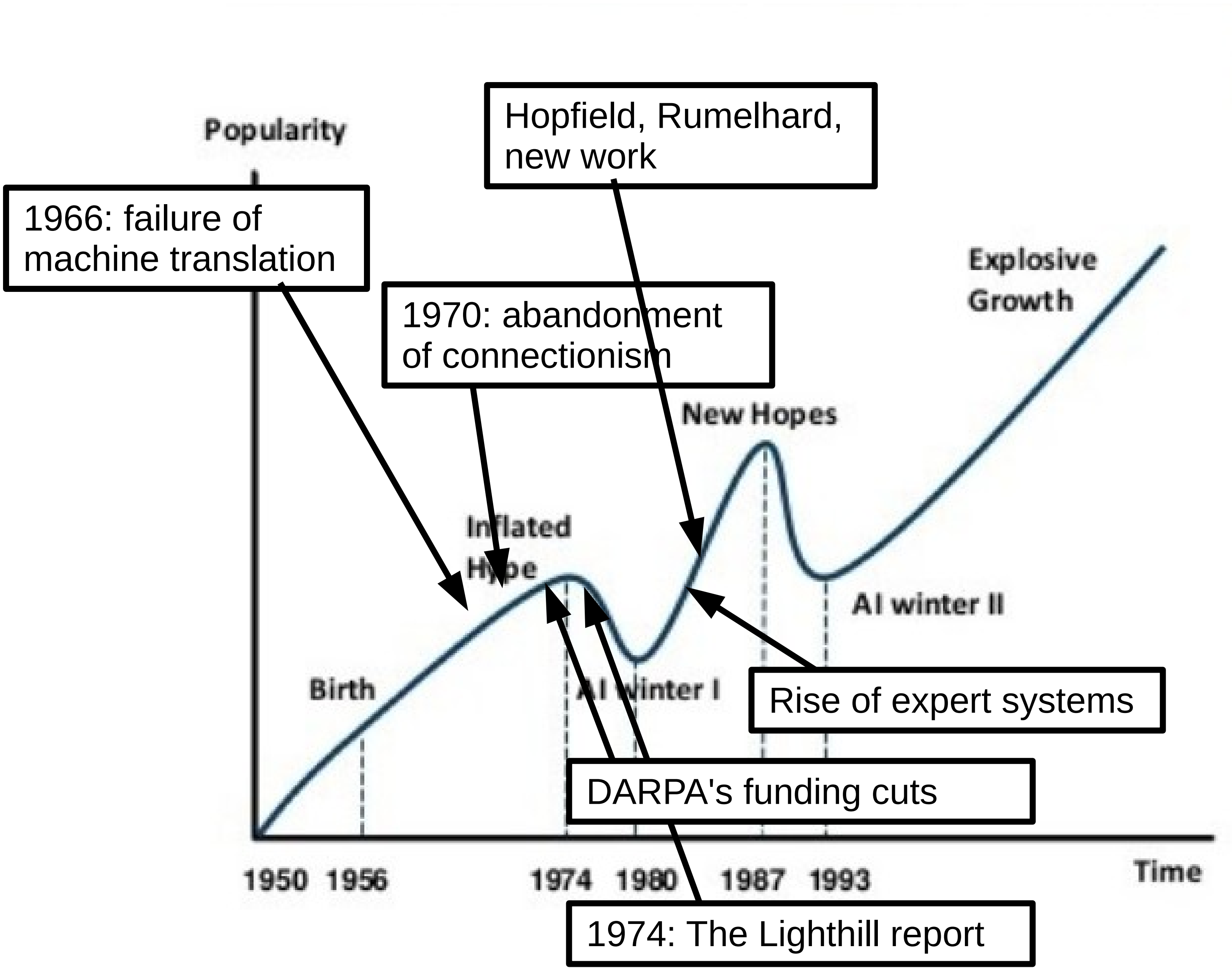
Computer: THE BOX.

Person: How many blocks are not in the box?

Computer: FOUR OF THEM.



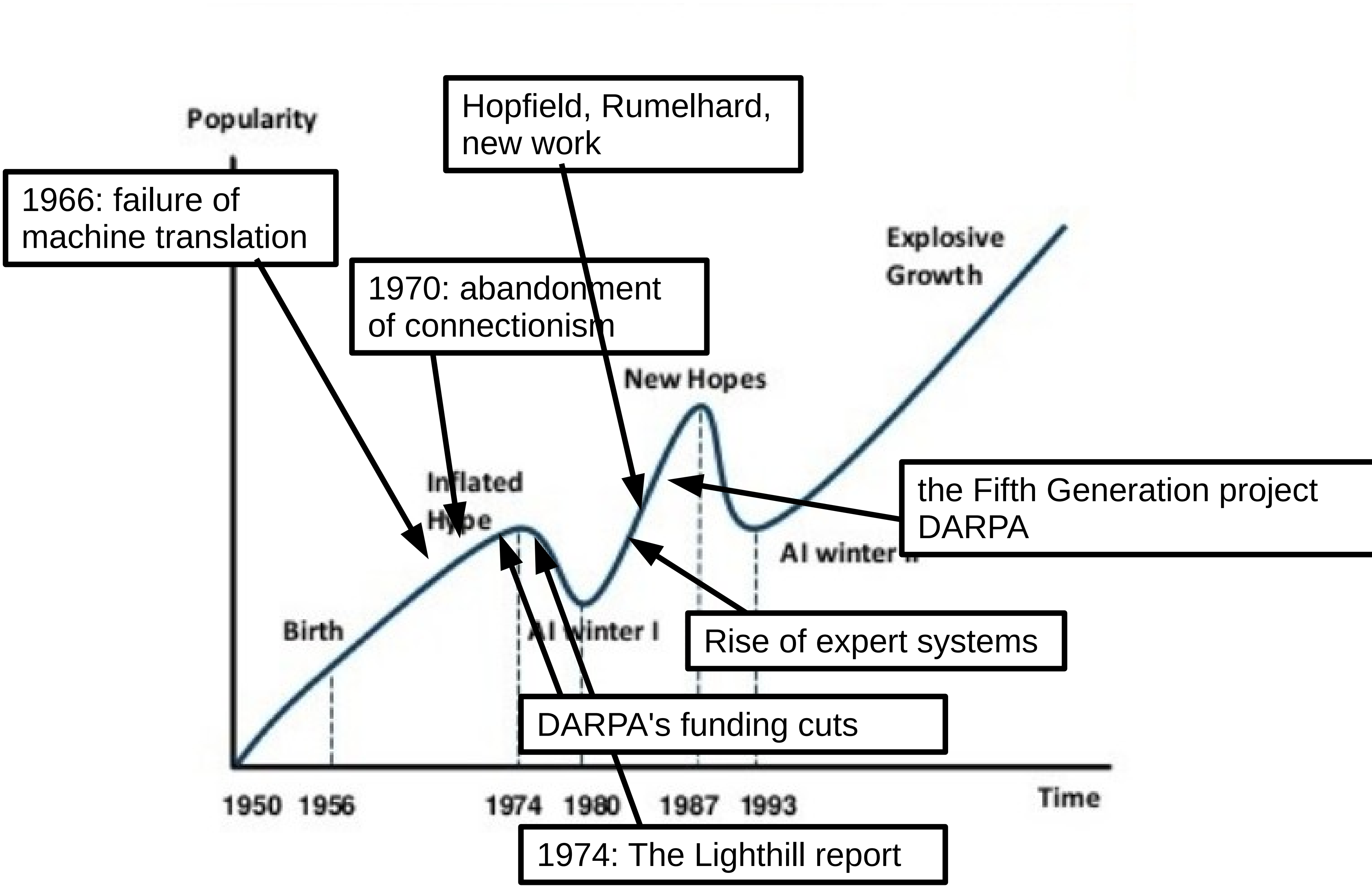
AI hypes, winters,

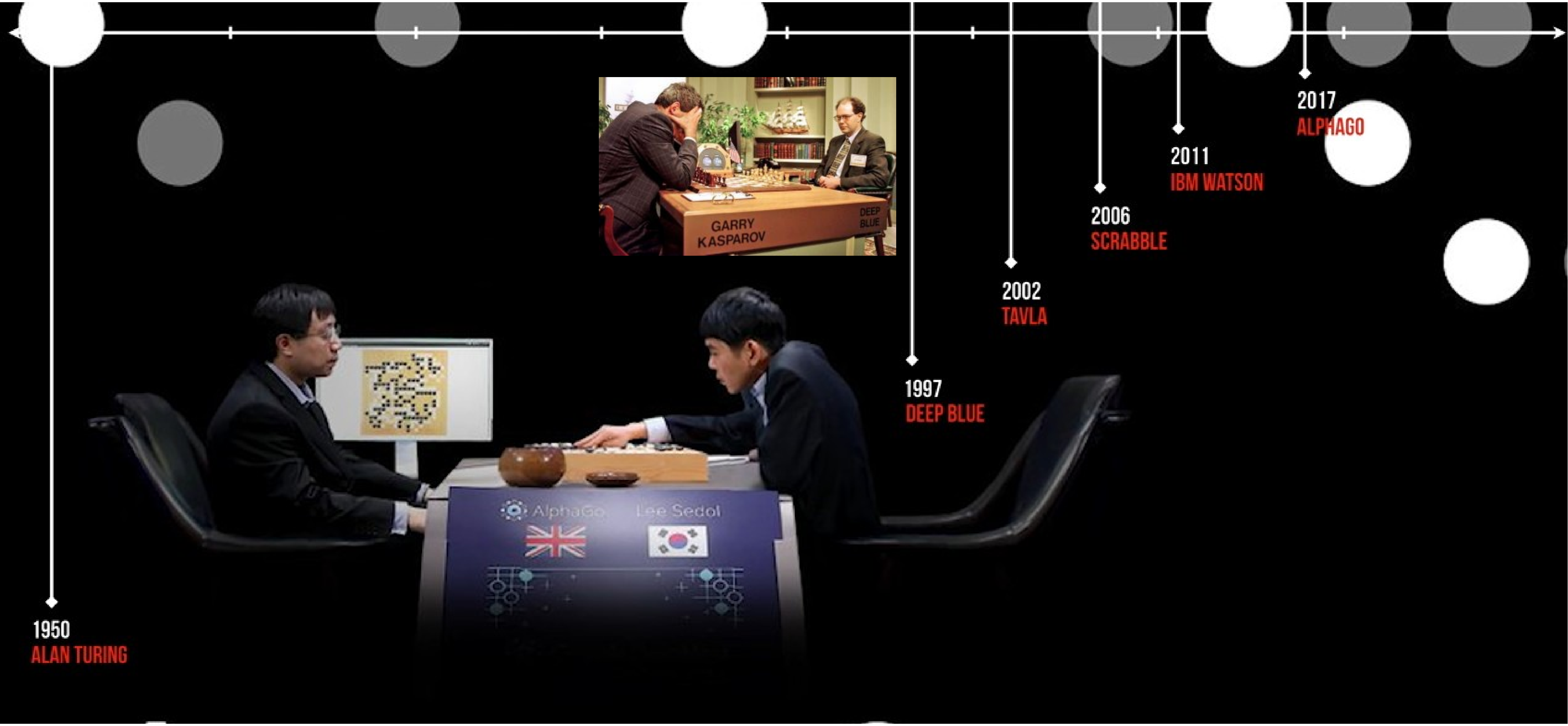


A dose of reality (1966-74)

- Simple syntactic manipulation did not scale
  - ELIZA (example rule: if sentence contains “mother”, then say: “tell me more about your family”)
- Intractability
- Machine evolution did not scale
- *Perceptrons* by Minsky and Paper book in 1969 with negative result on representation capability of 1-layer ANNs

AI hypes, winters,

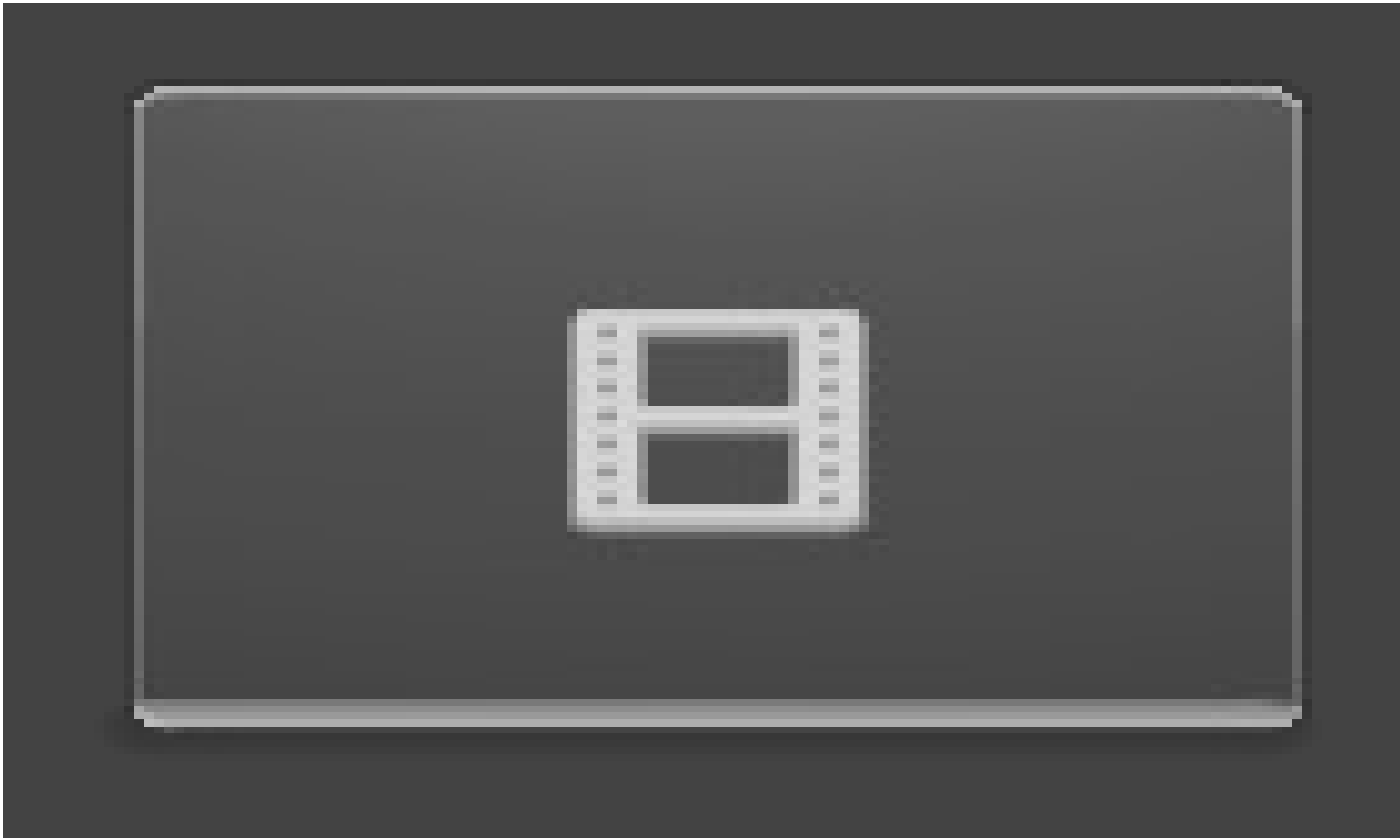




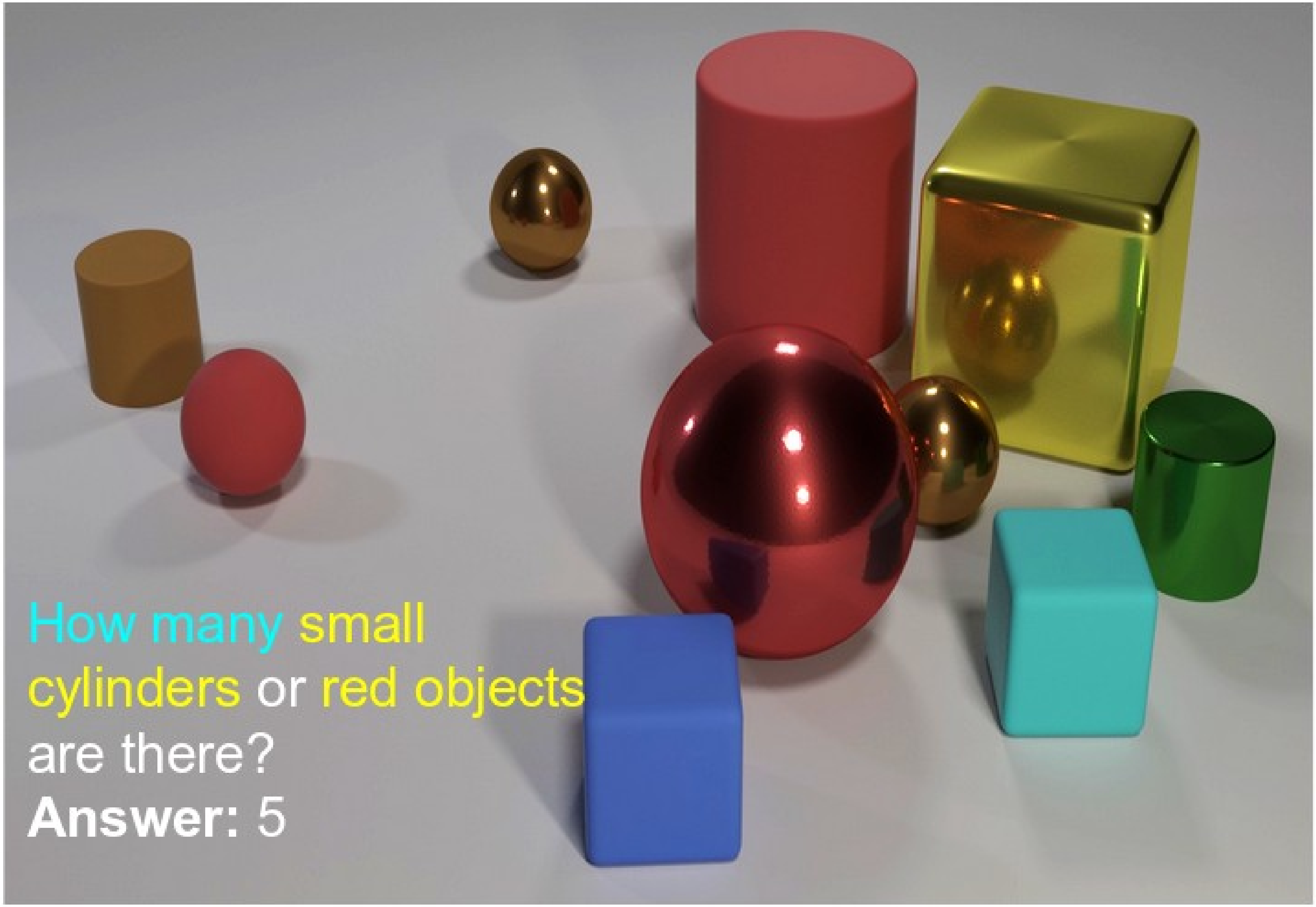
1950
ALAN TURING
GARRY
KASPAROV
DEEP
BLUE
1997
DEEP BLUE
2002
TAVLA
2006
SCRABBLE
2011
IBM WATSON
2017
ALPHAGO
AlphaGo
Lee Sedol

20 vs 361

Deepmind - Reinforcement Learning



State-of-the-art in Natural Language Processing and Scene Recognition

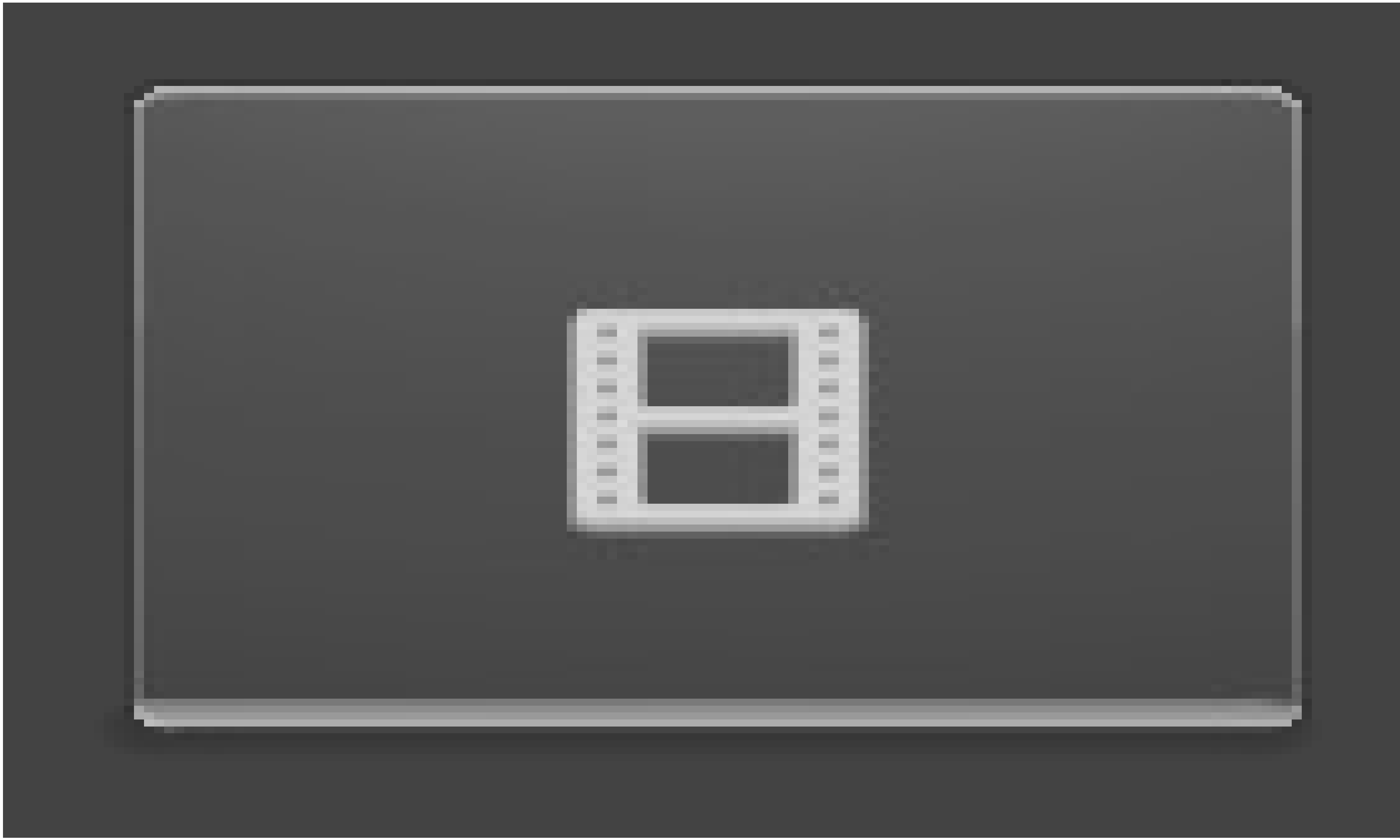


*adapted from Dr. Aykut Erdem

State-of-the-art in robotics

State-of-the-art in robotics

State-of-the-art in robotics



What is AI?

Views of AI fall into four categories (Humanly vs. Rationally)

- Acting rationally, Thinking humanly, Acting humanly, Thinking rationally,

``The exciting new effort to make computers think ... machines with minds, in the full and literal sense'' (Haugeland, 1985)  ``The automation of activities that we associate with human thinking, activities such as decision-making, problem solving, learning ...'' (Bellman, 1978)	``The study of mental faculties through the use of computational models'' (Charniak and McDermott, 1985)  ``The study of the computations that make it possible to perceive, reason, and act'' (Winston, 1992)
``The art of creating machines that perform functions that require intelligence when performed by people'' (Kurzweil, 1990)  ``The study of how to make computers do things at which, at the moment, people are better'' (Rich and Knight, 1991)	``A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes'' (Schalkoff, 1990)  `` AI … is concerned with intelligent behavior in artifacts.” (Nillson, 1998)

Acting rationally: rational agent

- Rational behavior: doing the right thing.
  - How do you define right thing?
  - The right thing: that which is expected to maximize <u>expected</u> goal achievement or outcome, given the available information
- Doesn't necessarily involve thinking, e.g.?
  - e.g., blinking reflex – but thinking should be in the service of rational action

Rational agents

- An agent is an entity that perceives and acts
- This course is about designing rational agents
- Abstractly, an agent is a function from percept histories to actions:
  - [f: $P^* \rightarrow A$]
- For any given class of environments and tasks, we seek the agent (or class of agents) with the best performance
- Caveat: computational limitations make perfect rationality unachievable
  - Design best program for given machine resources

State-of-the-art

Real-world interactions
Creativity

- ✓ Play a decent game of table tennis
- ✓ Drive safely along a curving mountain road
- ✗ Drive safely along Minibüs Caddesi
- ✓ Buy a week’s worth of groceries on the web
- ✓ Play a decent game of bridge
- ✗ Discover and prove a new mathematical theorem
- ✗ Design and execute a research program in molecular biology
- ✗ Write an intentionally funny story
- ✓ Give competent legal advice in a specialized area of law
- ✓ Translate spoken English into spoken Swedish in real time
- ✗ Converse successfully with another person for an hour
- ✗ Perform a complex surgical operation
- ✓ Clean the floors without further guidance
- ✗ Unload any dishwasher and put everything away

Chapter 2:
Intelligent Agents

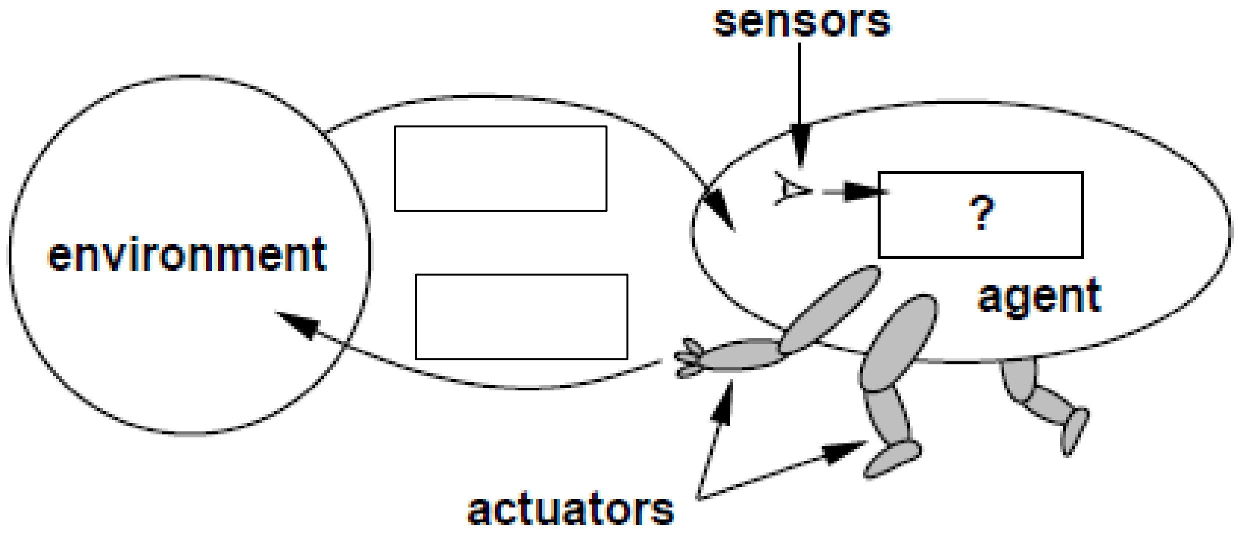
Overview

- PEAS (Performance, Environment, Actuators, Sensors)
- Environment types
- Agent functions and properties
- Agent types

What is an 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

Agents



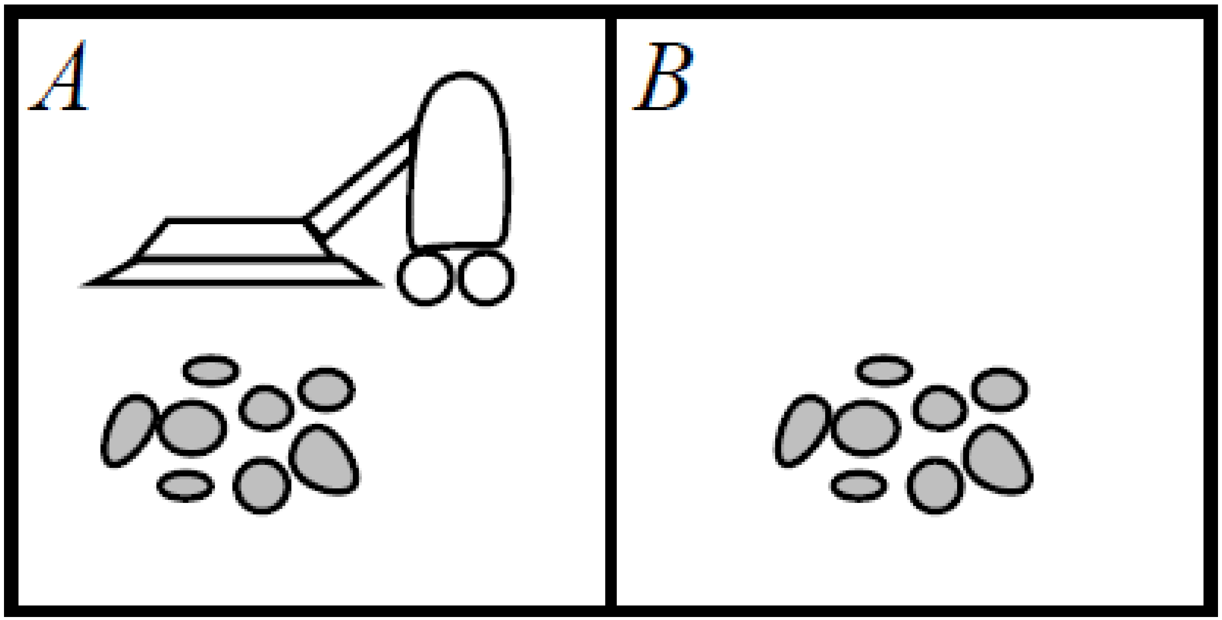
Agents include humans, robots, softbots, thermostats, etc.

The **agent function** maps from to

$f: P^* \rightarrow A$

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*

Simplest reflex agent?

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?

Rationality

- A rational agent is one that does the **right thing**.
- More precisely, what is rational at any given time depends on four things:
  - The performance measure that defines the criterion of success.
  - The agent's prior knowledge of the environment.
  - The actions that the agent can perform.
  - The agent's percept sequence to date.

Performance measure

- 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.
- Design measures according to what you want (as a behavior) and not according to what you think the agent should behave!
- Vacuum-cleaner?
  - Amount of dirt cleaned up.

Rationality

- Rationality maximizes expected performance while perfection maximizes actual performance.

Task Environment

- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent odometer, engine sensors, ?
- Consider, e.g., the task of designing an automated taxi driver ("beyond the capabilities of existing technology") :
- Performance measure:
  -
- Environment:
  -
- Actuators:
  -
- Sensors:
  -

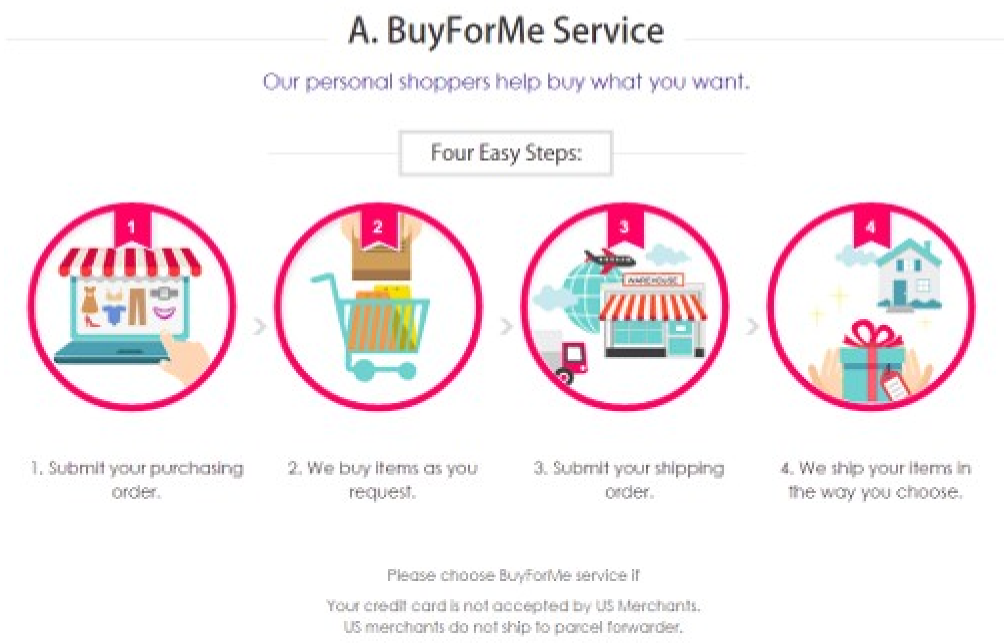
Task Environment

- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent odometer, engine sensors, ?
- Consider, e.g., the task of designing an automated taxi driver ("beyond the capabilities of existing technology") :
- Performance measure:
  - Safe, fast, legal, comfortable trip, maximize profits and ?
- Environment:
  - Roads, other traffic (?), pedestrians, customers, weather
- Actuators:
  - Steering wheel, accelerator, brake, signal, horn, ?
- Sensors:
  - Cameras, sonar, speedometer, GPS, odometer, engine sensors, ?

PEAS for Internet shopping agent

- Performance Measure?
- Environment?
- Actuators?
- Sensors?

PEAS for Internet shopping agent



- Performance Measure?
  - price, quality, appropriateness, efficiency
- Environment?
  - current and future WWW sites, vendors, shippers
- Actuators?
  - display to user, follow URL, fill in form
- Sensors?
  - HTML pages (text, graphics, scripts)

PEAS for Part-picking robot

- Performance measure:
  –
- Environment:
  –
- Actuators:
  –
- Sensors:
  –

PEAS for 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

Agent Characteristics

- Embodiment
- Situatedness
- Autonomy
- Adaptivity
- Sociability

Agent Characteristics

- **Situatedness:** The agent receives some form of sensory input from its environment, and it performs some action that changes its environment in some way. Examples of environments: the physical world and the Internet.
- **Embodiment:** Having a physical body
- **Autonomy:** The agent can act without direct intervention by humans or other agents and that it has control over its own actions and internal state.

Agent Characteristics

- **Adaptivity :** The agent is capable of
  - (1) reacting flexibly to changes in its environment
  - (2) taking goal directed initiative (i.e., is pro-active), when appropriate;
  - and (3) learning from its own experience, its environment, and interactions with others.
- **Sociability:** The agent is capable of interacting in a peer-to-peer manner with other agents or humans.

Environment Types – Categorize in different dimensions

- **Fully observable** vs. **partially observable**.
  -

- **Deterministic** vs. **stochastic**.
  -
  -
  -
- **Episodic** vs. **sequential**.
  -

Environment Types – Categorize in different dimensions

- **Fully observable** vs. **partially observable**.
  - If an agent's sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable. Example?
- **Deterministic** vs. **stochastic**.
  - Guaranteed effect.
  - 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. Example? Board games? Chess, backgammon
- **Episodic** vs. **sequential**.
  - In an episodic task environment, the agent's experience is divided into atomic "episodes."
  - Episode: Single cycle of an agent perceiving and taking an action
  - Episodic: If the choice depends on the current episode and not on previous episodes
    - Easier to operate.
  - In sequential environments, the current decision may affect all future decisions. Examples? Chess?

Environment Types

- **Static** vs. **dynamic**.

- **Discrete** vs. **continuous**.
  -

- **Single-agent** vs. **multi-agent**.
  -

Environment Types

- **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.
  - If the environment itself does not change with the passage of time but the agent's **performance score** does, then we say the environment is semidynamic. Example?
- **Discrete** vs. **continuous**.
  - The discrete/continuous distinction can be applied to the *state* of the environment, to the way *time* is handled, and to the *percepts* and *actions* of the agent. Example?
  - Possible to convert continuous environments into discrete environments (with loss of precision)
- **Single-agent** vs. **multi-agent**.
  - Taxi driver? Whether B's behavior is best described as maximizing a performance measure depending on A's performance measure
  - Competitive
  - Cooperative

Environment types

	Backgammon	Internet Shopping	Taxi
Observable?			
Deterministic?			
Episodic?			
Static?			
Discrete?			
Single-agent?			

Environment types

	Backgammon	Internet Shopping	Taxi
Observable?	Yes	No	No
Deterministic?	No	Partly	No
Episodic?	No	No	No
Static?	Yes	Semi	No
Discrete?	Yes	Yes	No
Single-agent?	No	Yes (except auctions)	No

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

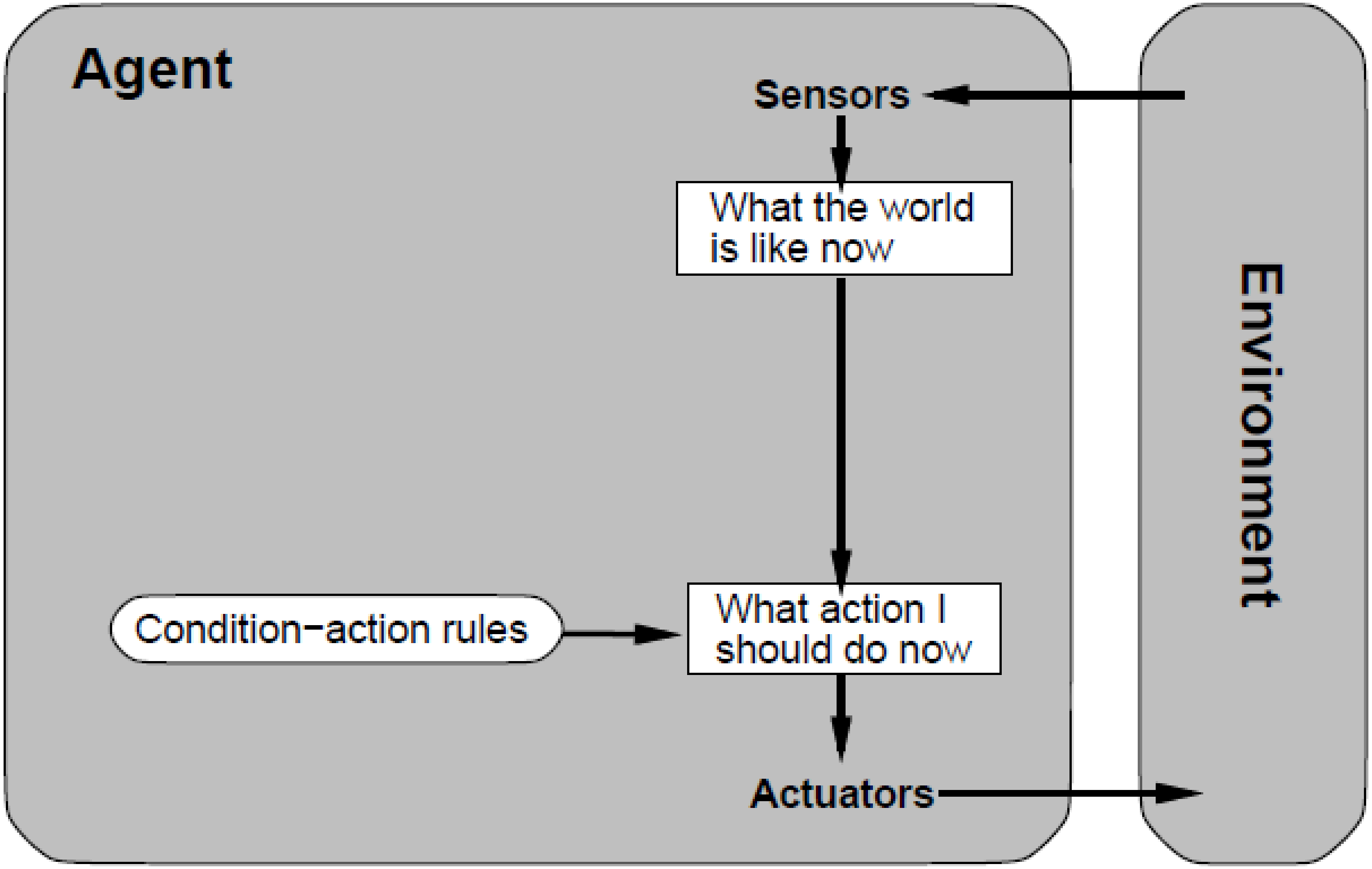
Structure of agent

- **Agent = architecture + program**
- **Agent Function:**
  - Mathematically speaking, we say that an agent's behavior is described by the agent function that maps any given percept sequence to an action.
- **Agent Program:**
  - Take the current percept as input and return action.
  - The implementation of the agent function for an artificial agent is called the agent program.
  - Why only current percept?
    - Only information provided by the environment.

Agent types

- Four basic agent 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

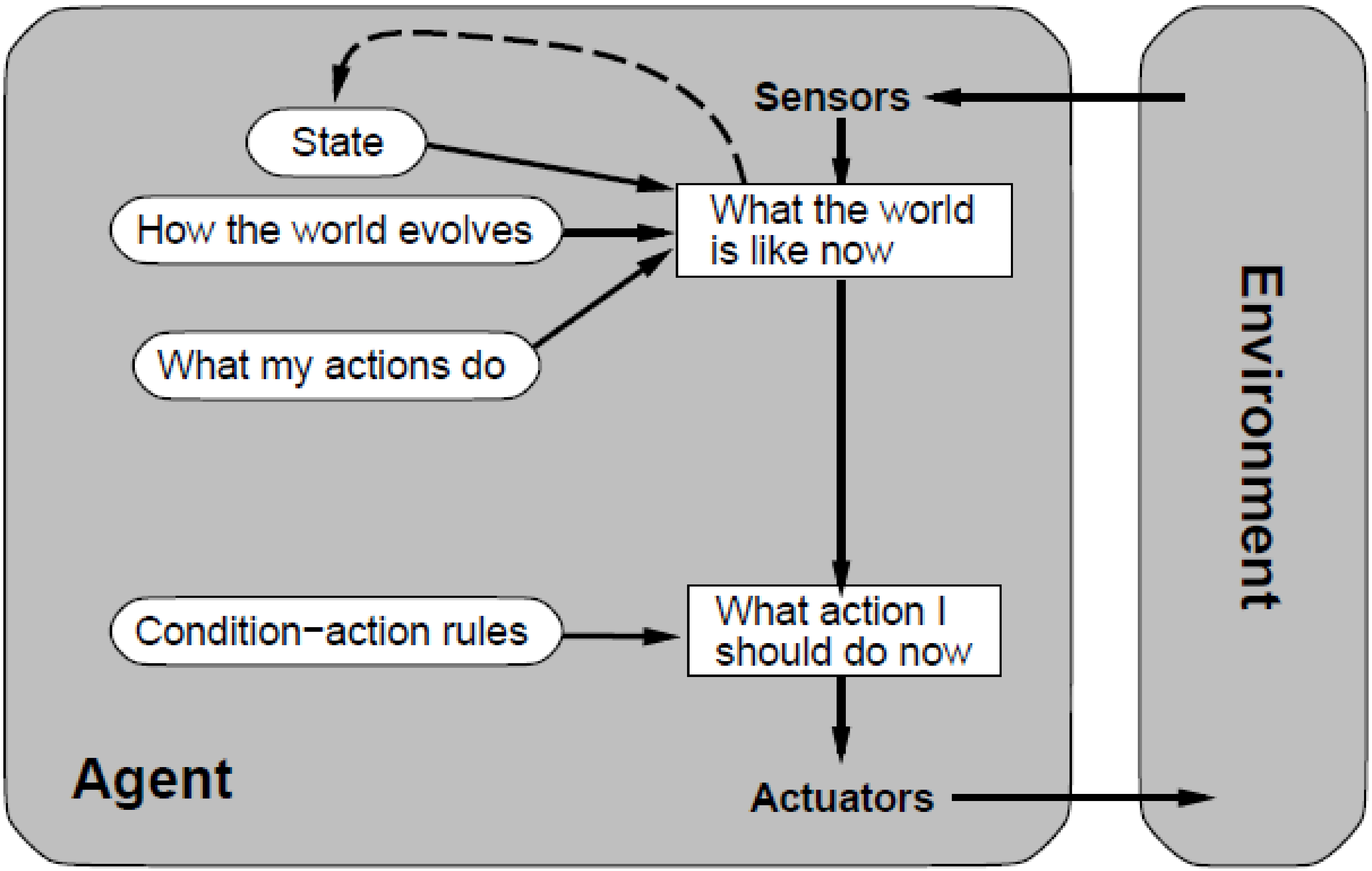
Simple Reflex Agents



Simple Reflex Agents

- Select actions based only on the current percept, ignoring the rest of the percept history.
- Table lookup of percept-action pairs defining all possible condition-action rules necessary to interact in an environment
- • Problems
  - Possible condition-action rules too big to generate and to store (Chess has about 10^{120} states, for example)
  - No knowledge of non-perceptual parts of the current state
    - Get into loops - randomize
  - Not adaptive to changes in the environment; requires entire table to be updated if changes occur

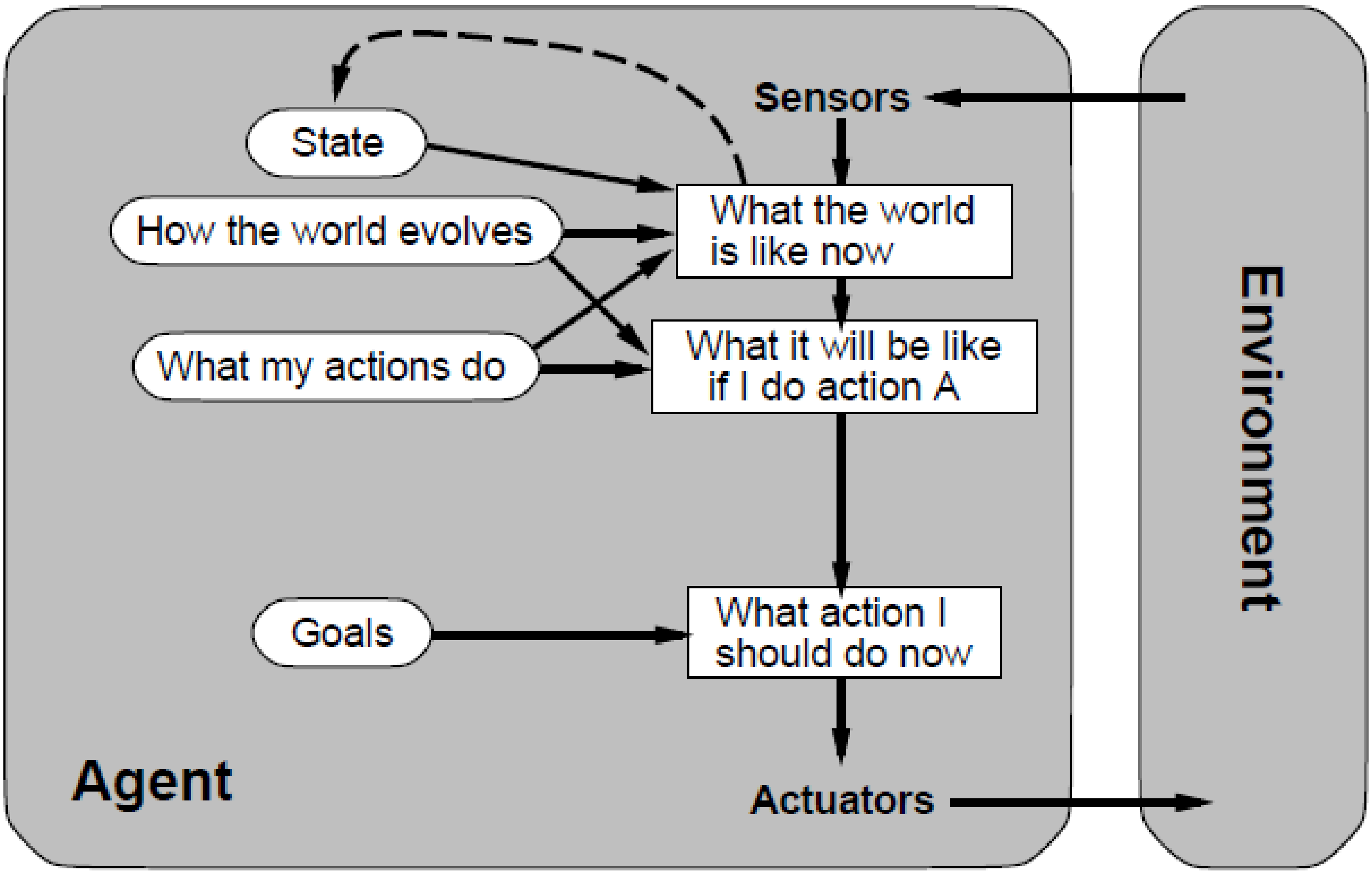
Reflex Agents with state



Reflex Agent with State Model-based reflex agents

- The knowledge about “how the world works” is called a model of the world.
- An agent that uses such a model is called a model-based agent.
- Encode "internal state" of the world to remember the past as contained in earlier percepts
- **Needed** because sensors do not usually give the entire state of the world at each input (what did we call this environment?), so perception of the environment is captured over time.
- Requires ability to represent change in the world; one possibility is to represent just the latest state, but then can't reason about hypothetical courses of action
  - Ability to “model” how world evolves independent of agent – other driver
  - Ability to “model” how world evolves as a result of agent actions

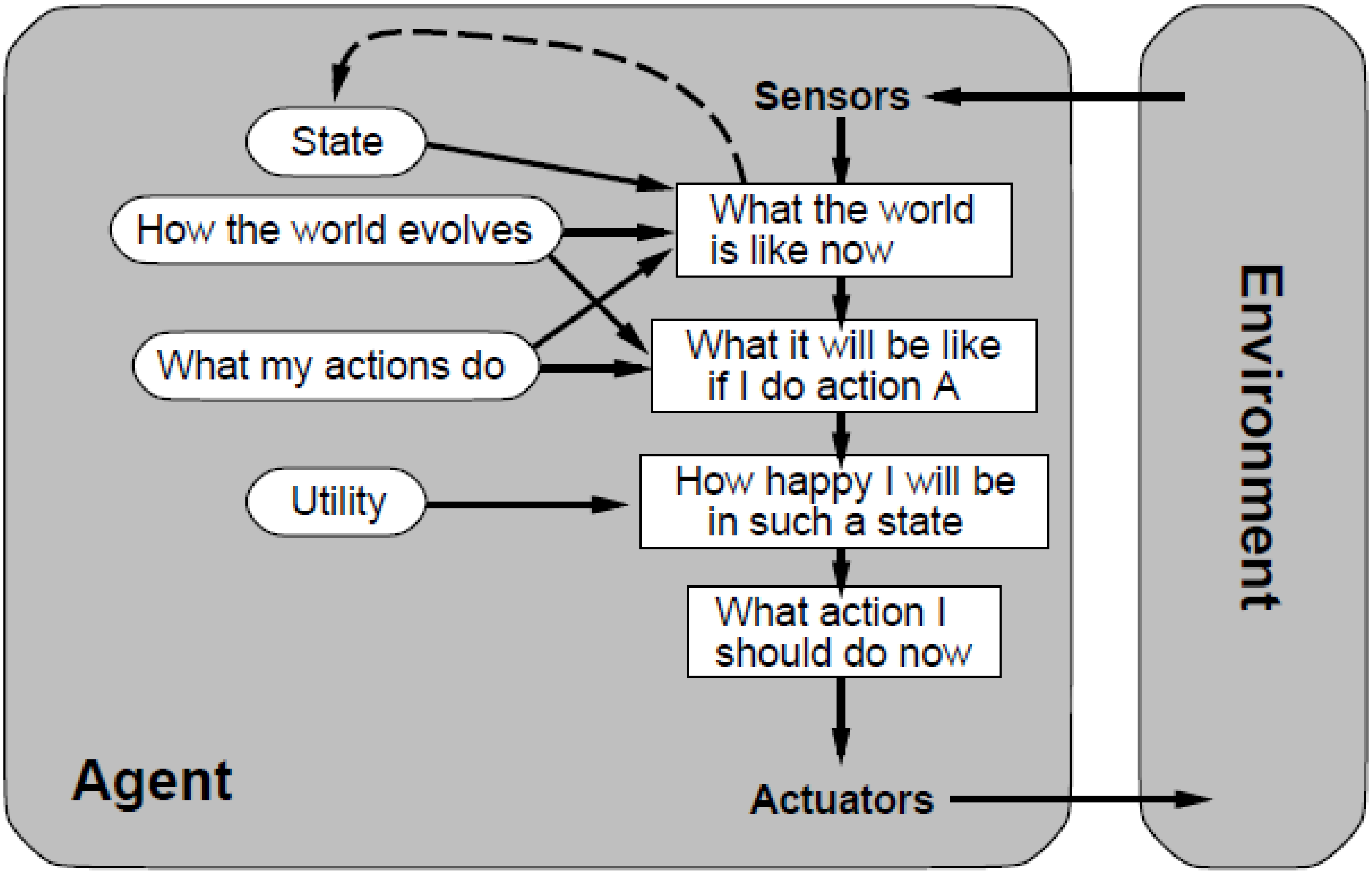
Goal-based Agent



Goal-based Agents

- A taxi in a junction, turn left or right?
- Choose actions so as to achieve a (given or computed) goal = a description of a desirable situation
- Keeping track of the current state is often not enough --- need to add goals to decide which situations are good
  - Think about a **goal** state where the desired goal holds
  - **Plan** or **search** a sequence of actions such that applying those actions will transform the current state into the goal state
  - Combine goal info and possible next states.
- Deliberative instead of reactive. Not reflex anymore.
- May have to consider long sequences of possible actions before deciding if goal is achieved --- involves consideration of the future, "what will happen if I do...?"
- Goal-based agents are more flexible – no need to rewrite lookup tables.

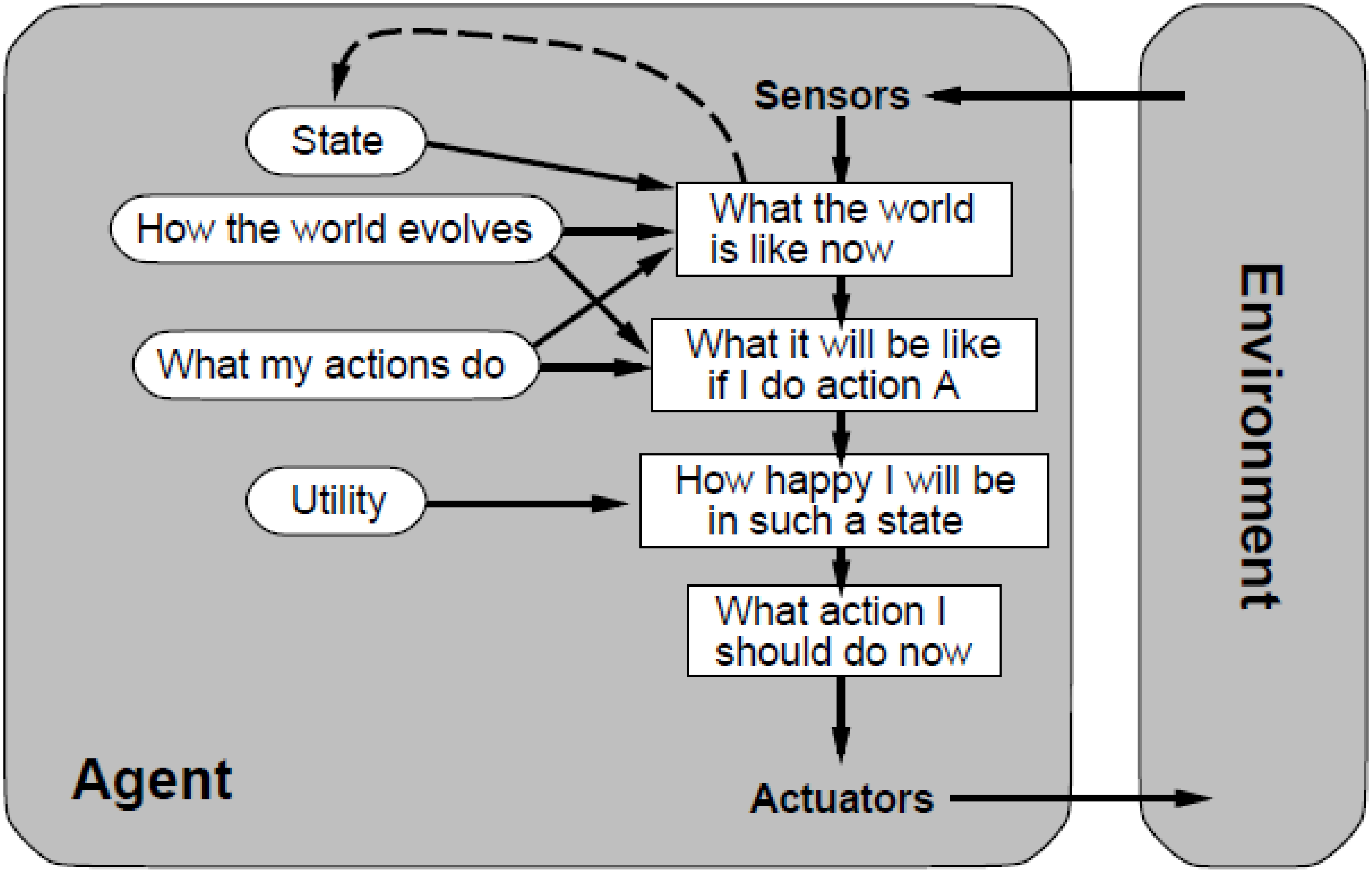
Utility-based Agents



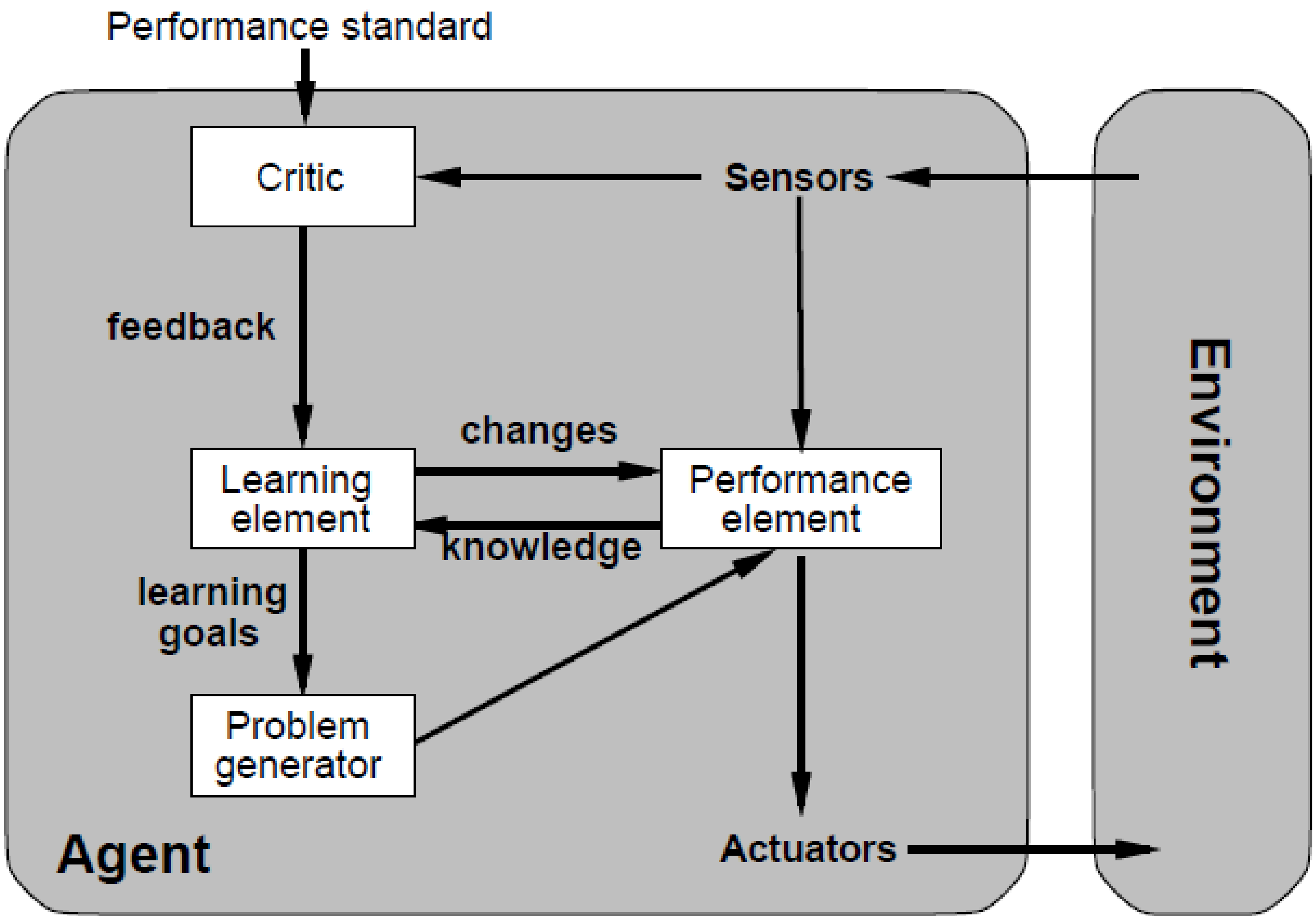
Utility-based Agents

- When there are multiple possible alternatives, how to decide which one is best?
- A goal specifies a crude distinction between a happy and unhappy state, but often need a more general performance measure that describes "degree of happiness"
- **Goals alone are not really enough.**
  - Taxi: different sequences to the same goal are safer, quicker, more reliable or cheaper.
- **Utility function** U: State → Reals
  - indicates a measure of success or happiness when at a given state
  - A state has higher utility if it is preferred over another.
  - Allows decisions comparing choice between conflicting goals, and choice between likelihood of success and importance of goal (if achievement is uncertain)

What is missing?



Learning Agents



Learning agents - components

- **learning element**, which is responsible for making improvements,
- **performance element**, which is responsible for selecting external actions. The performance element is the entire agent: it takes in percepts and decides on actions.
- **critic** gives feedback from the on how the agent is doing with respect to a fixed performance standard.
- **problem generator** is responsible for suggesting actions that will lead to new and informative experiences.

Summary

- Agents interact with environments through actuators and sensors
- The agent function describes what the agent does in all circumstances
- The performance measure evaluates the environment sequence
- A perfectly rational agent maximizes expected performance

Summary

- Agent programs implement (some) agent functions
- PEAS descriptions define task environments
- Environments are categorized along several dimensions:
  - Observable? Deterministic? Episodic? Static? Discrete? Single-agent?
- Several basic agent architectures exist:
  - Reflex, reflex with state, goal-based, utility-based
  - Learning vs. non-learning