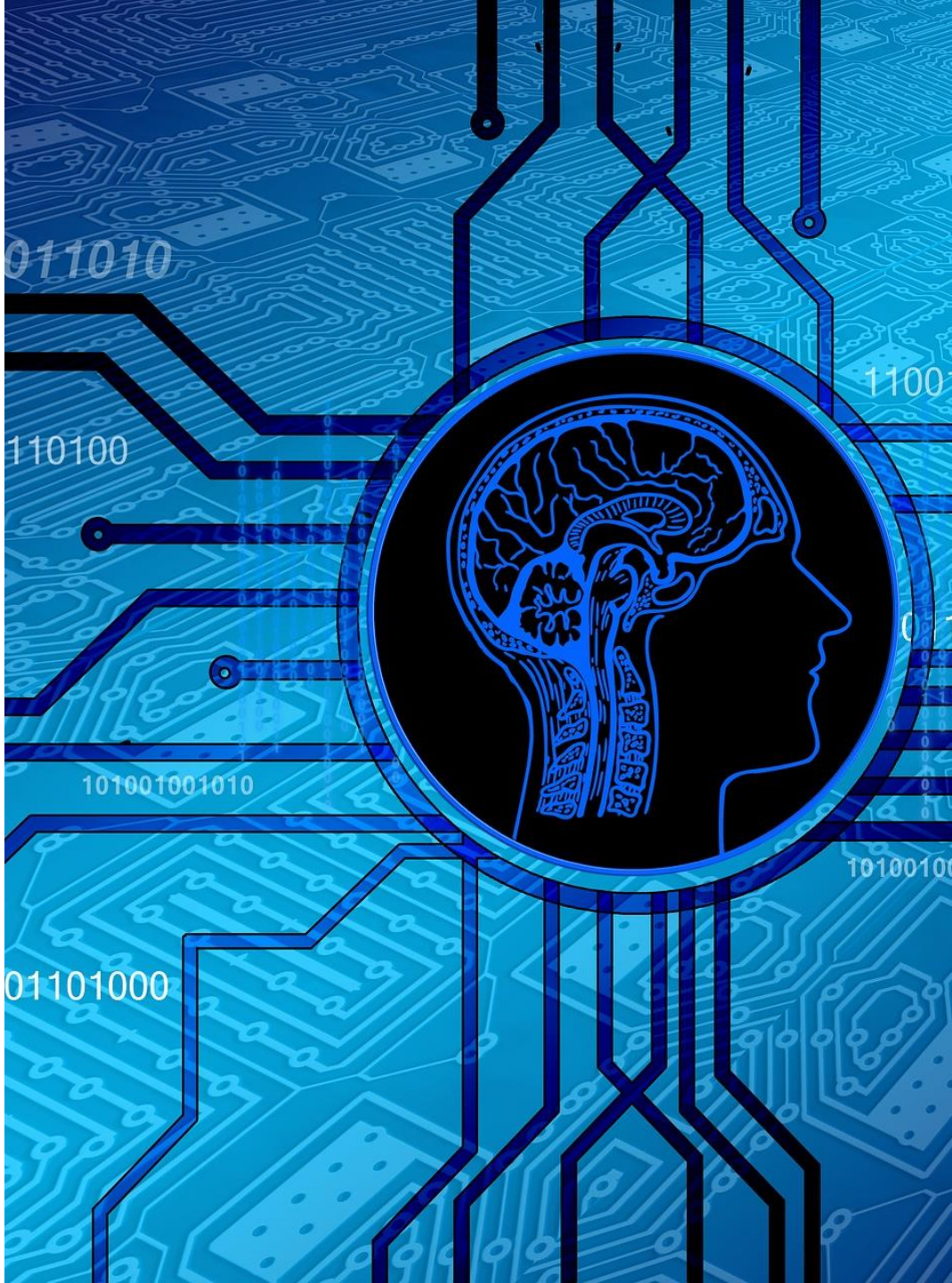
Intelligent Agents and their Environments

Petros Papapanagiotou

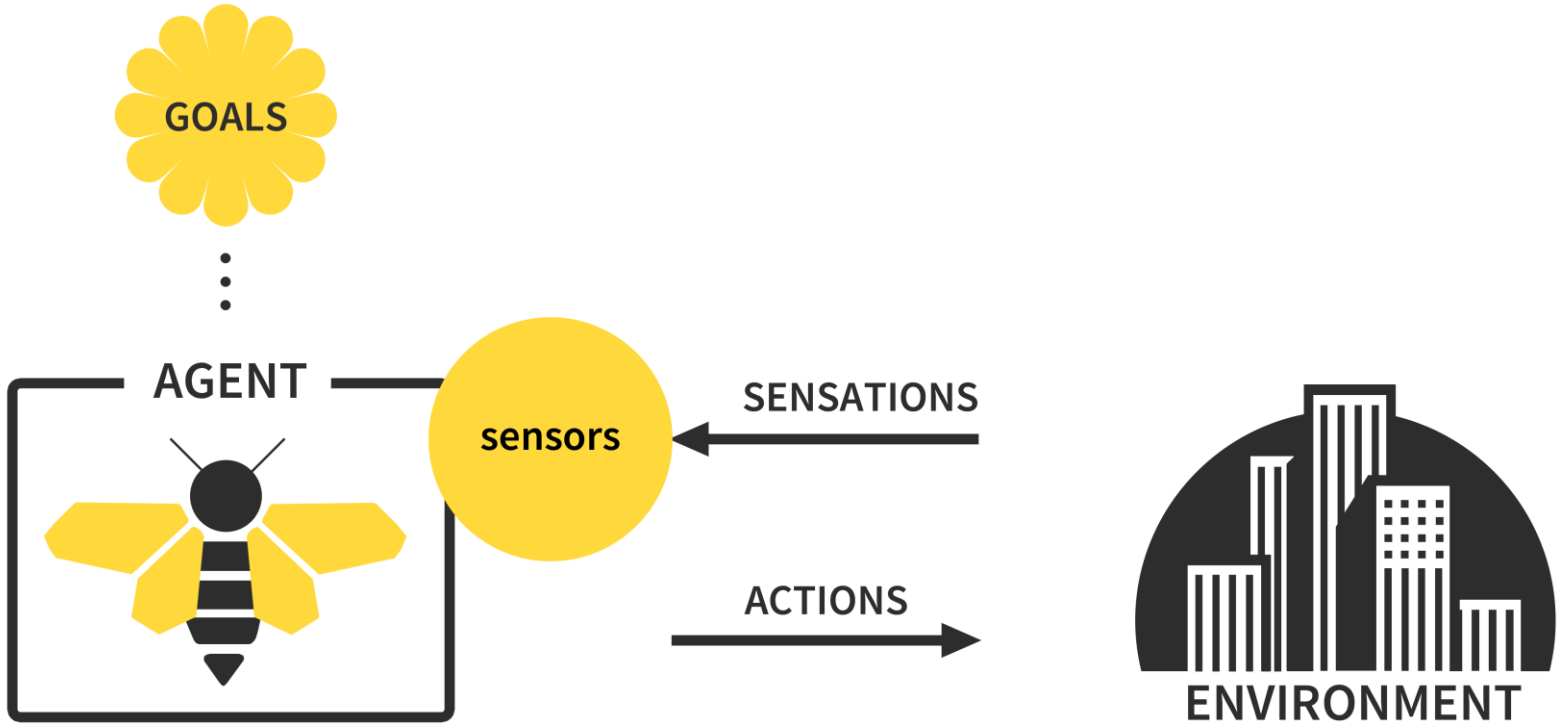
Informatics 2D: Reasoning and Agents

Lecture 1

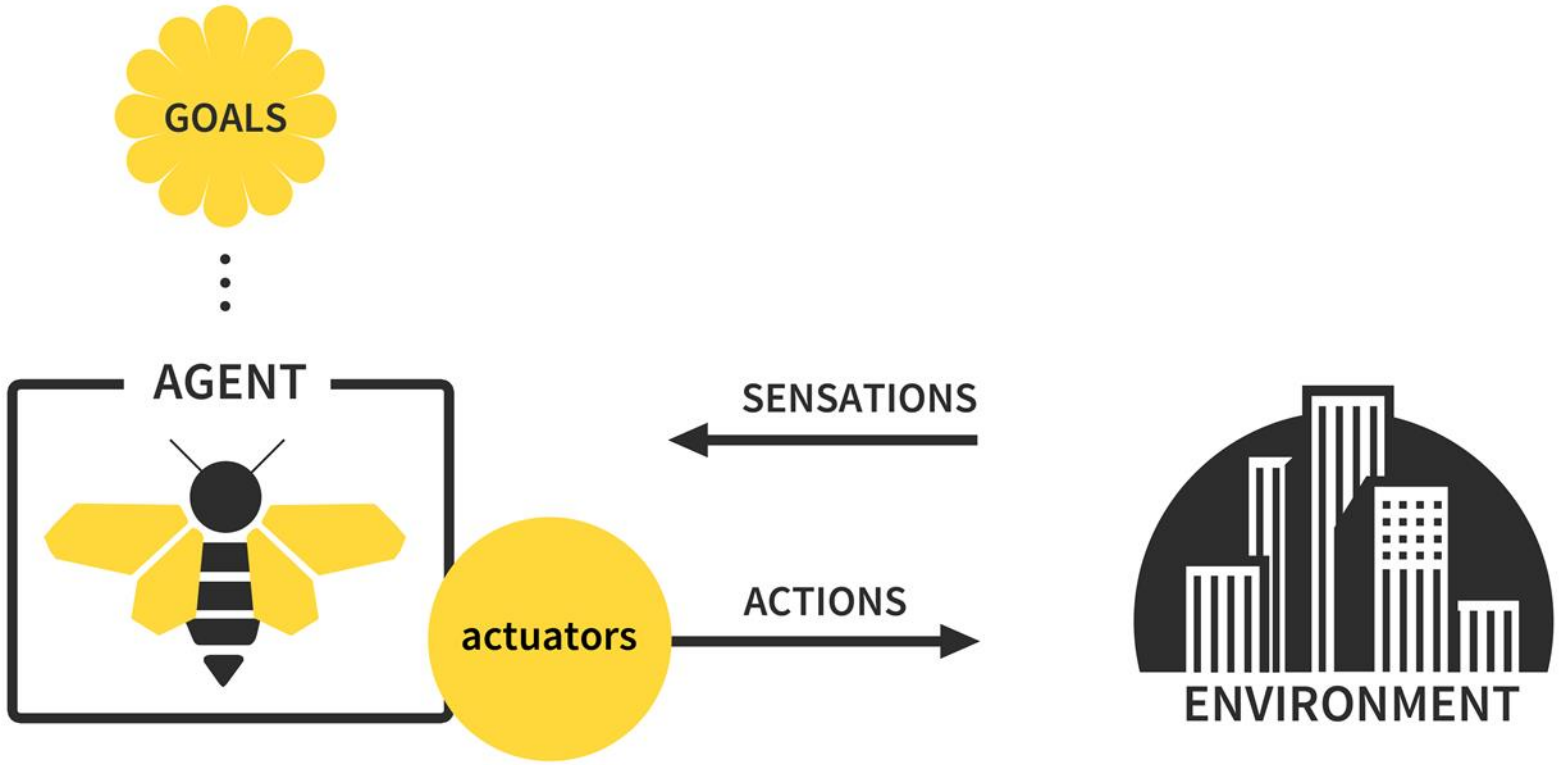


What is an Intelligent Agent?

Structure



Structure



An agent

Perceives its *environment*,

Through its *sensors*,

Then achieves its *goals*,

By acting on its environment via *actuators*.

Categorise agents

Environment

Goals

Percepts

Actions

Example: Mail sorting

Conveyor belt of letters

Route letter into correct bin

Array of pixel intensities

Route letter into bin

Smart Home |

INF2D: REASONING AND AGENTS

Example: Smart home

occupants enter and leave house, occupants enter and leave rooms; daily variation in outside light and temperature

occupants warm, room lights are on when room is occupied, house energy efficient

signals from temperature sensors, movement sensors, clock, sound sensor, weather sensor

room heaters on/off, lights on/off

Example: Autonomous car

streets, other vehicles, pedestrians, traffic signals/lights/signs

safe, fast, legal trip

camera, GPS signals, speedometer, sonar

steer, accelerate, brake

Type of Intelligent Agents

Simple Reflex Agents

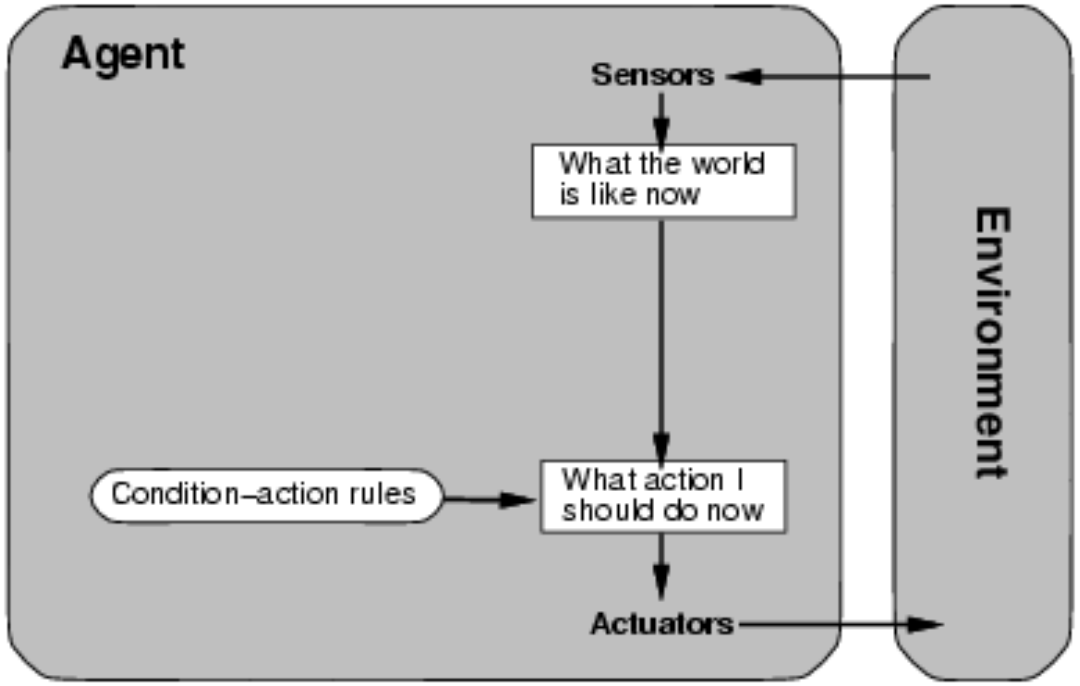
Action depends only on immediate percepts.

Implement by *condition-action rules*.

Example:

- Agent: Mail sorting robot
- Environment: Conveyor belt of letters
- Rule: e.g. *city=Edinburgh → put in Scotland bag*

Simple Reflex Agents



function SIMPLE-REFLEX-AGENT(*percept*)
returns *action*
  persistent: *rules* (set of condition-action rules)
    state ← INTERPRET-INPUT(*percept*)
    rule ← RULE-MATCH(*state*, *rules*)
    action ← rule.ACTION
return *action*

Model-Based Reflex Agents

Action may depend on history or unperceived aspects of the world.

Need to maintain *internal world model*.

Example:

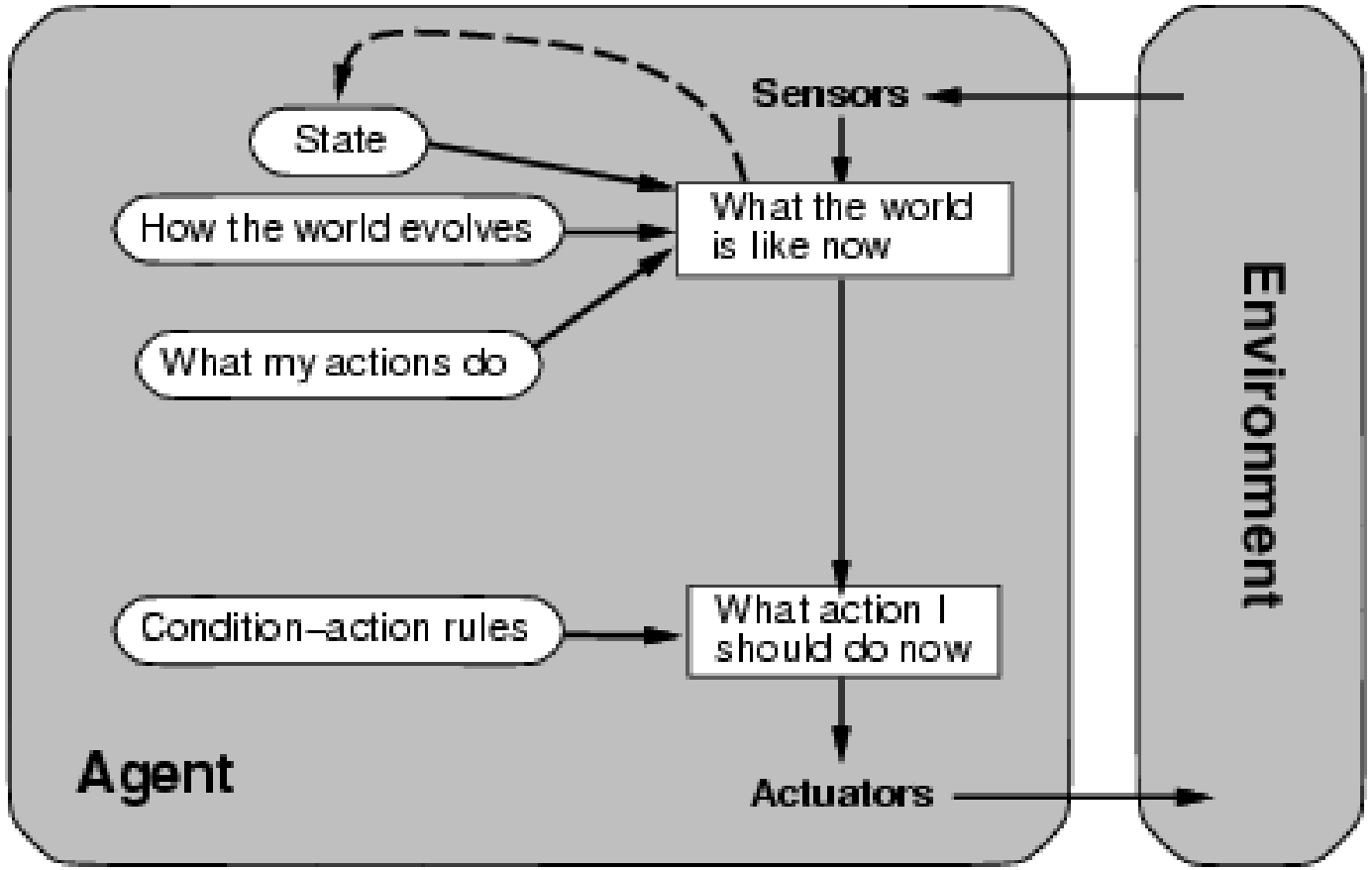
Agent: robot vacuum cleaner

Environment: dirty room, furniture

Model: map of room, which areas already cleaned

Sensor/model tradeoff.

Model-Based Reflex Agents



function REFLEX-AGENT-WITH-STATE(*percept*)

returns *action*

persistent: *state*, description of current world state

model, description of how the next state depends on current state and action

rules, a set of condition-action rules

action, the most recent action, initially none

state ← UPDATE-STATE(*state, action, percept, model*)

rule ← RULE-MATCH(*state, rules*)

action ← rule.ACTION

return *action*

Goal-Based Agents

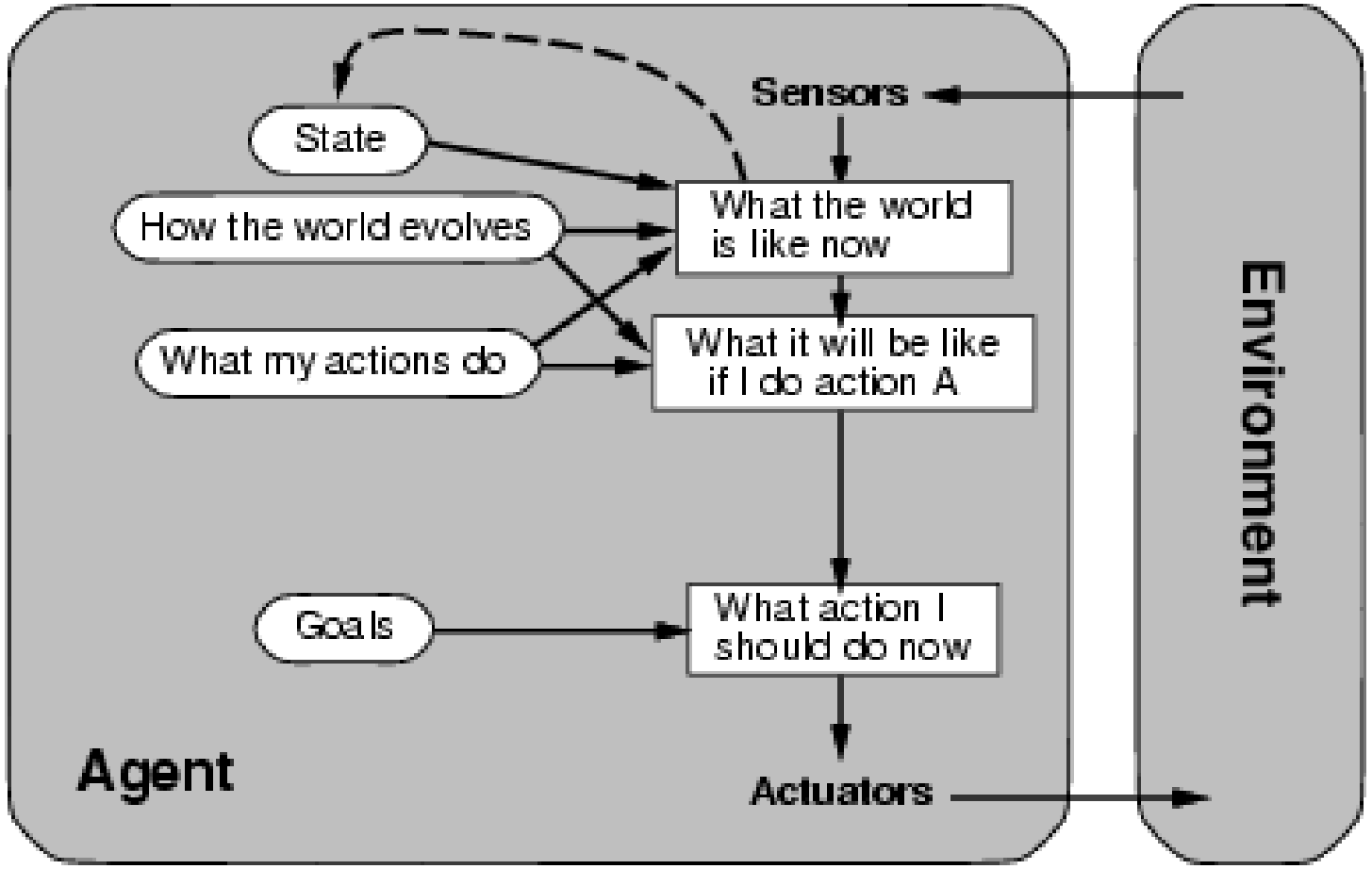
Agents so far have fixed, implicit goals.

We want agents with variable goals.

Forming plans to achieve goals is a topic for later.

Example:

- Agent: robot maid
- Environment: house & people.
- Goals: clean clothes, tidy room, table laid, etc



Goal-Based Agents

Utility-Based Agents

Agents so far have had a single goal.

Agents may have to juggle conflicting goals.

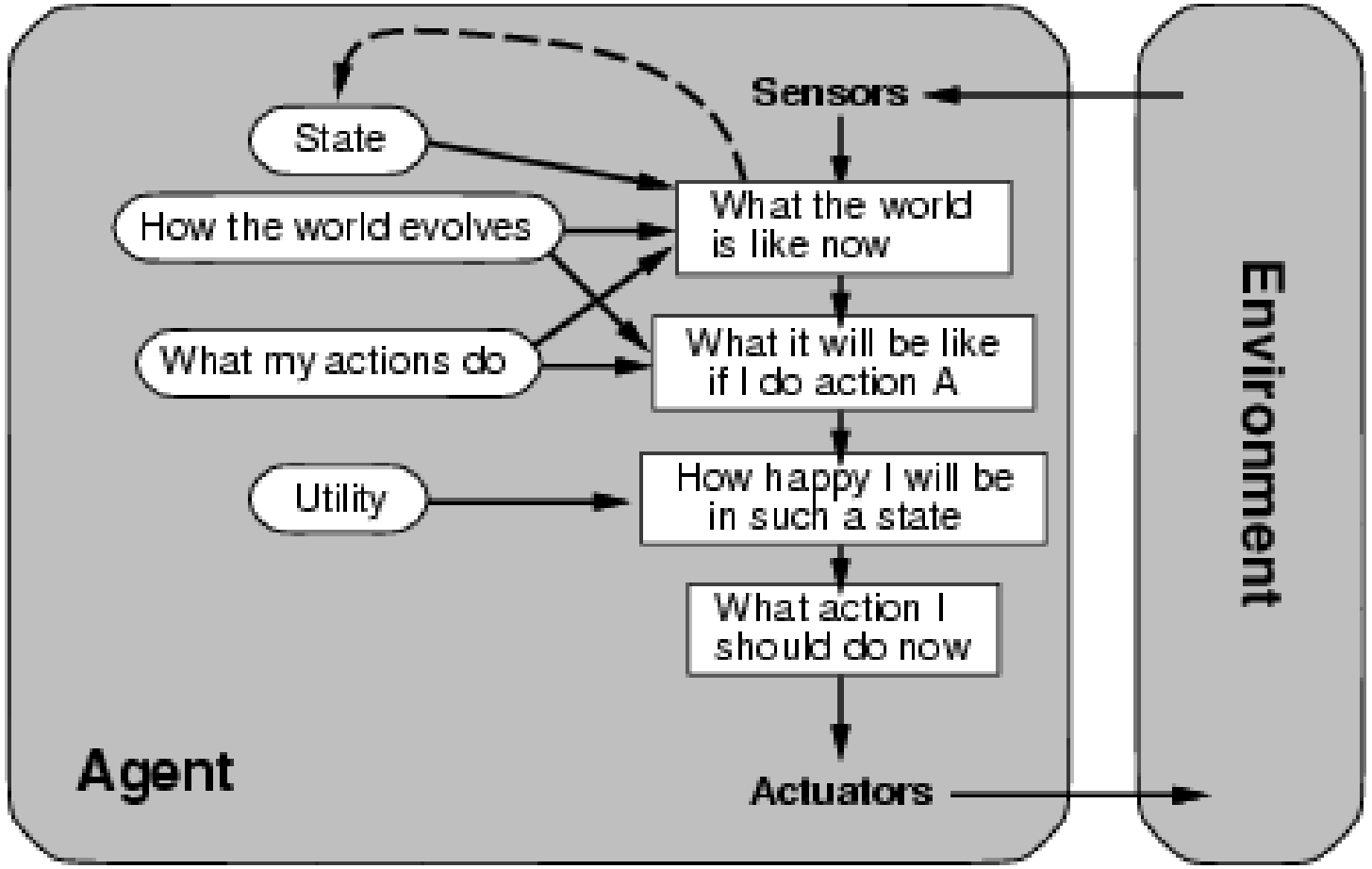
Need to optimise utility over a range of goals.

Utility: measure of *goodness* (a real number).

Combine with probability of success to get *expected utility*.

Example:

- Agent: autonomous car.
- Environment: roads, vehicles, signs, etc.
- Goals: stay safe, reach destination, be quick, obey law, save fuel, etc.



Utility-Based Agents

Learning Agents

How do agents improve their performance in the light of experience?

- Generate problems which will test performance.
- Perform activities according to rules, goals, model, utilities, etc.
- Monitor performance and identify non-optimal activity.
- Identify and implement improvements

We will not be covering learning agents, but this topic is dealt with in several honours-level courses (see also R&N, Ch. 18-21).

Exercise

Consider a chess playing program.

What sort of agent would it need to be?

Solution

Simple-reflex agent: but some actions require some memory (e.g. castling)

Model-based reflex agent: but needs to reason about future

Goal-based agent: but only has one goal

Utility-based agent: might consider multiple goals with limited lookahead

Learning agent: learns from experience or self-play

Types of Environments

Observable?

FULLY

Source

PARTIALLY

Source

Observable?

FULLY

PARTIALLY

agent's sensors describe environment fully

vs.

some parts of the environment not visible; noisy sensors

Source

Source

Deterministic?

DETERMINISTIC

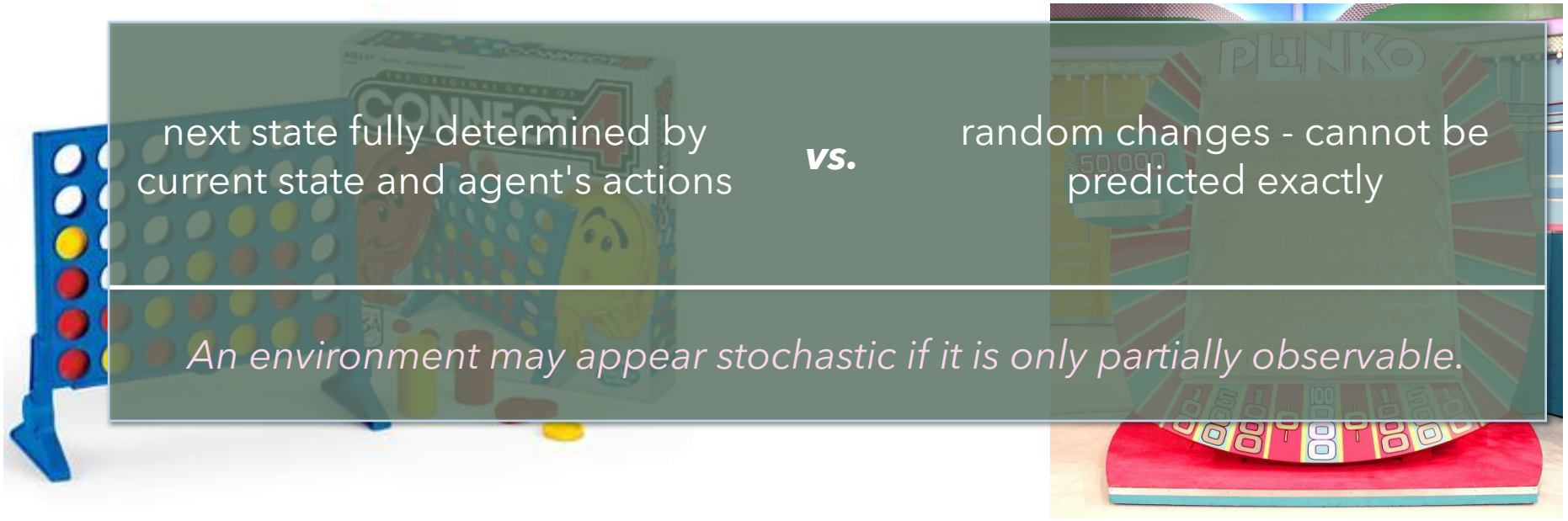
STOCHASTIC

Source

Deterministic?

DETERMINISTIC

STOCHASTIC



next state fully determined by current state and agent's actions

vs.

random changes - cannot be predicted exactly

An environment may appear stochastic if it is only partially observable.

Source

Sequential?

EPISODIC

[Source]

SEQUENTIAL

Sequential?

EPISODIC

SEQUENTIAL

next episode does not depend on previous actions

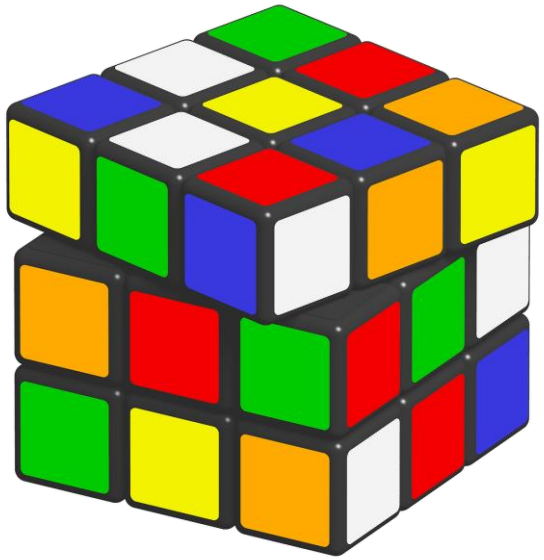
vs.

actions affect the future

Source

Static?

STATIC



DYNAMIC

Source

Static?

STATIC

DYNAMIC

environment unchanged while agent deliberates

vs.

environment can change at any time or even continuously

[Source]

Discrete?

DISCRETE

Source

CONTINUOUS

Source

Discrete?

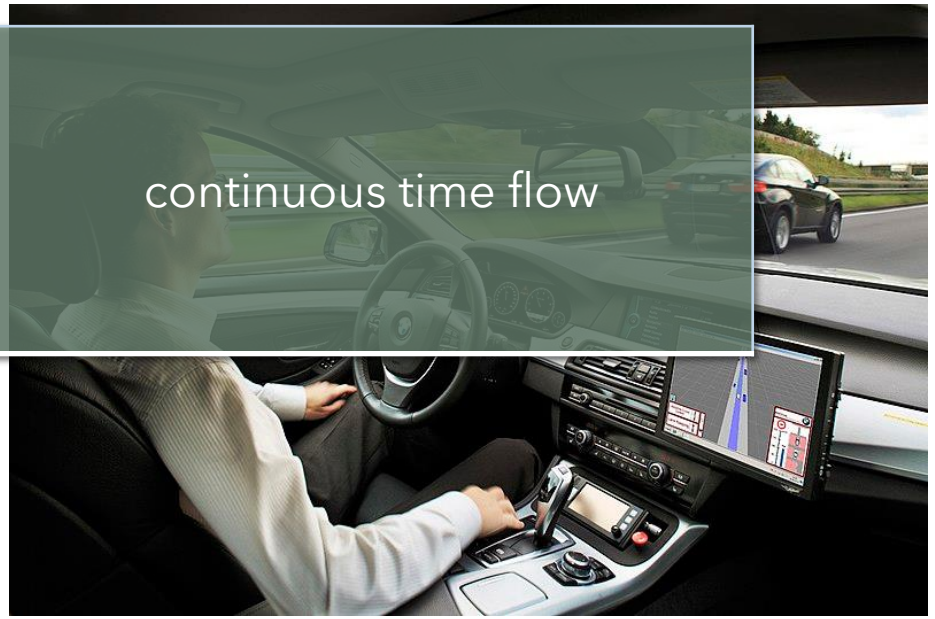
DISCRETE

CONTINUOUS

percepts, actions and episodes are discrete

vs.

continuous time flow

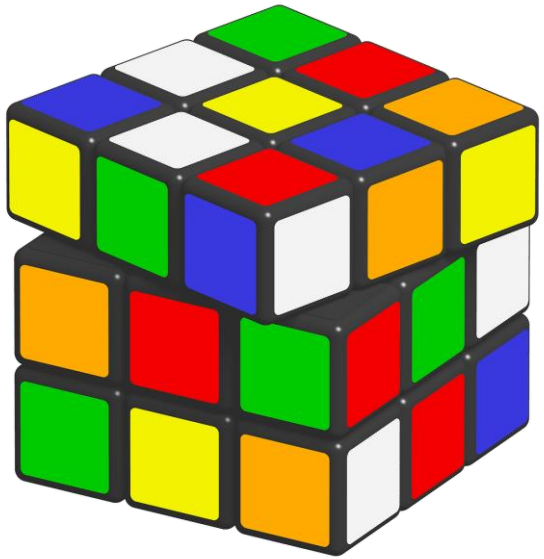


[Source](#)

[Source](#)

How many agents?

SINGLE



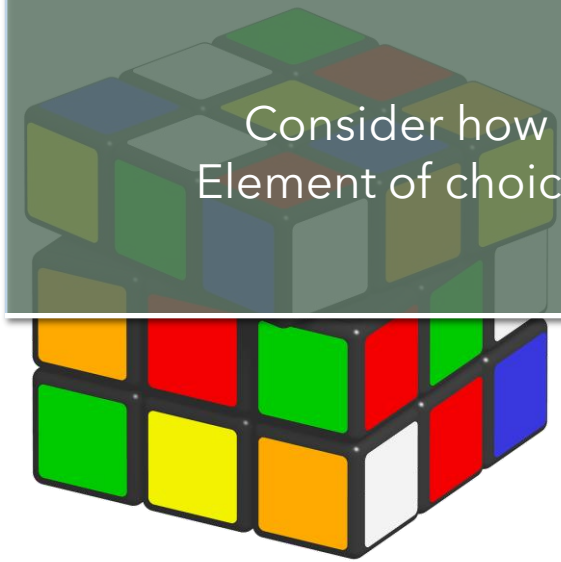
MULTI-AGENT

Source

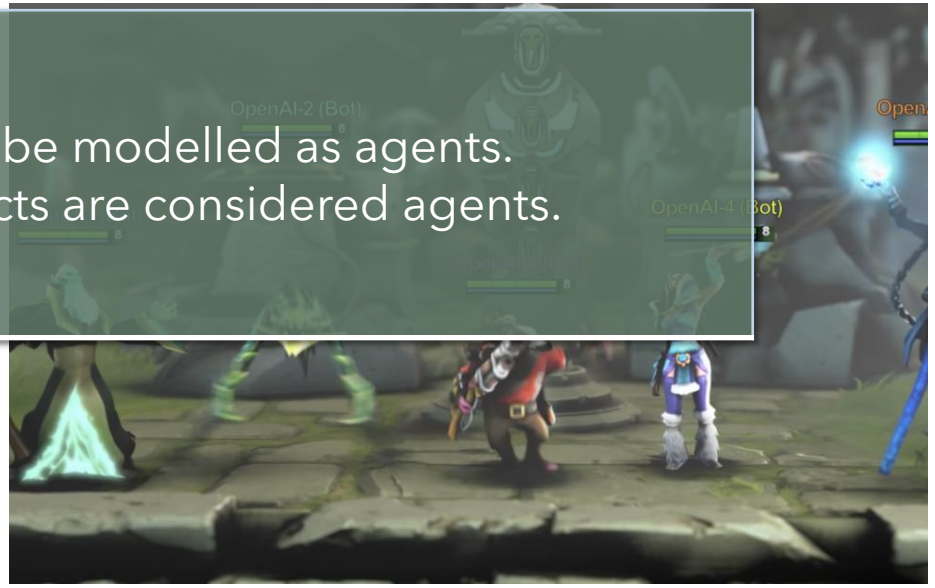
How many agents?

SINGLE

MULTI-AGENT



Consider how many object must be modelled as agents.
Element of choice over which objects are considered agents.



Source

Summary

An agent might have any combination of these properties:

- from "**benign**": i.e., fully observable, deterministic, episodic, static, discrete and single agent
- to "**chaotic**": partially observable, stochastic, sequential, dynamic, continuous and multi-agent

What are the properties of the environment that would be experienced by

- a mail-sorting robot?
- a smart home?
- a car-driving robot?

Why?

- Understanding what is an agent and how it can be modelled.
- Understanding the environment and the assumptions or considerations that need to be made.
- Making the right design decisions and choosing the right tools.
- Managing the complexity.