Introduction to Computational Intelligence

Lecture 2

Outline

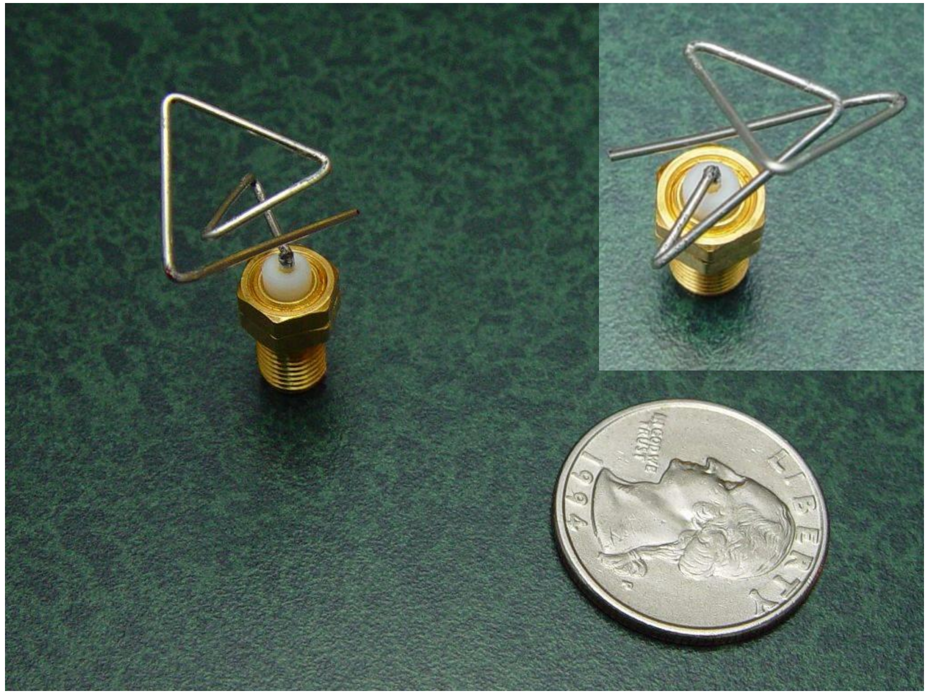
- Fundamental Definitions
  - Computational Intelligence Paradigms
    - ✓ ANN
    - ✓ Fuzzy systems
    - ✓ Evolutionary computation
    - ✓ Genetic algorithm
- Example Applications
- Summary
- Pop quiz

Basic definitions Cont.

- **Evolutionary Computation (EC):**
  - **Source of inspiration** - biological evolution and <u>natural selection</u>.
  - **Model** - solves optimization problems by generating, evaluating and modifying a population of possible solutions.
  - **Extensions** –
    - ✓ Genetic algorithms (GA), evolutionary programming (EP),
    - ✓ Evolution strategies, genetic programming, swarm intelligence and optimization,
    - ✓ Differential evolution, evolvable hardware,
    - ✓ Multi-objective optimization
    - ✓ etc.

Basic definitions Cont.

- **Example application of EC** –
  - NASA's Evolutionary Antenna (Destination NASA - Evolutionary Antenna Synthesis - YouTube)
    - ✓ The complex shape of the antennas are designed by an EC-based design application to get the best radiation pattern.



Source: https://www.nasa.gov/

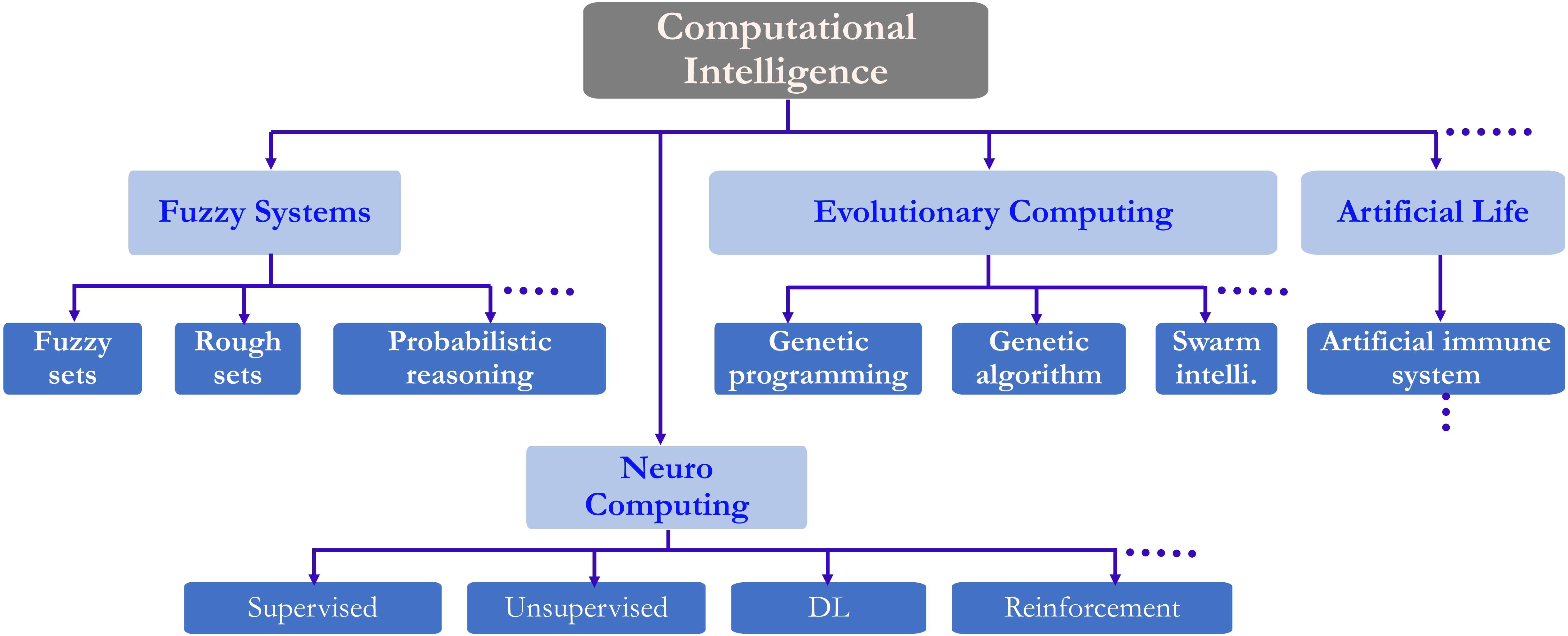
Basic definitions Cont.

- **Genetic algorithms (GA):**
- **Model** - incorporates natural evolution mechanisms, including crossover, mutation, and survival of the fittest.

- **EC vs GA**: Evolutionary computing algorithms are like genetic algorithms, but do not incorporate crossover.

- **Application**: They are more often used for optimization (search for the best), but also are used for classification.
  - Genetic algorithm tunes up public speakers (Genetic algorithm tunes up public speakers | New Scientist)

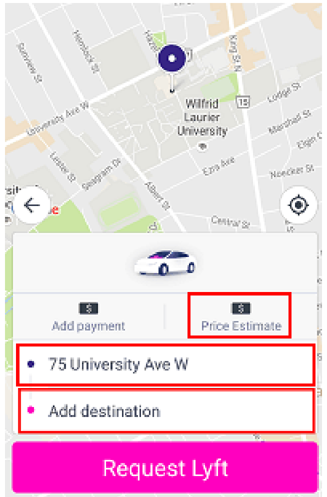
Computational Intelligence Paradigms

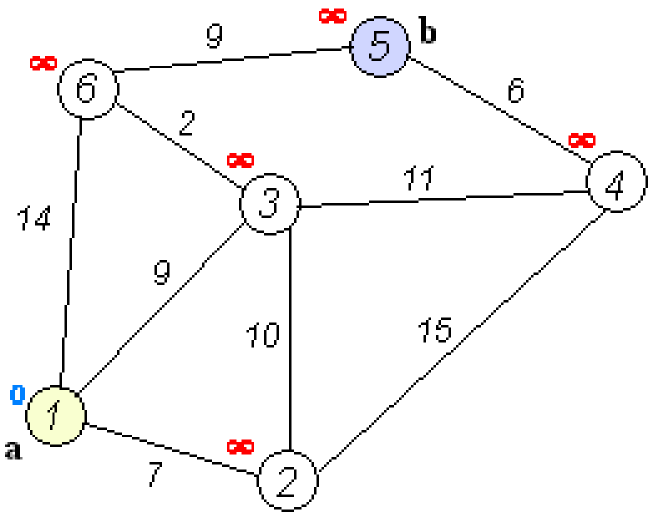


CI Example Applications

- **Google's AI-Powered Predictions:**
  - Using anonymized [location data from smartphones]()
  - Google Maps can analyze the speed of movement of traffic at any given time.

- **Ridesharing Apps Like Uber and Lyft:**
  - How do they determine the price of your ride?
  - How do they minimize the wait time once you hail a car?
  - How do these services optimally match you with other passengers to minimize detours?
  - The answer to all these questions is CI and ML





- **Plagiarism Checkers:**
  - Many high school and college students are familiar with services like *Turnitin*, a popular tool used by instructors to analyze students' writing for plagiarism.
  - While *Turnitin* doesn't reveal precisely how it detects plagiarism, research demonstrates how CI can be used to develop a plagiarism detector.

https://www.techemergence.com/everyday-examples-of-ai/ Image: Dijkstra's algorithm (Motherboard)

CI Example Applications Cont.

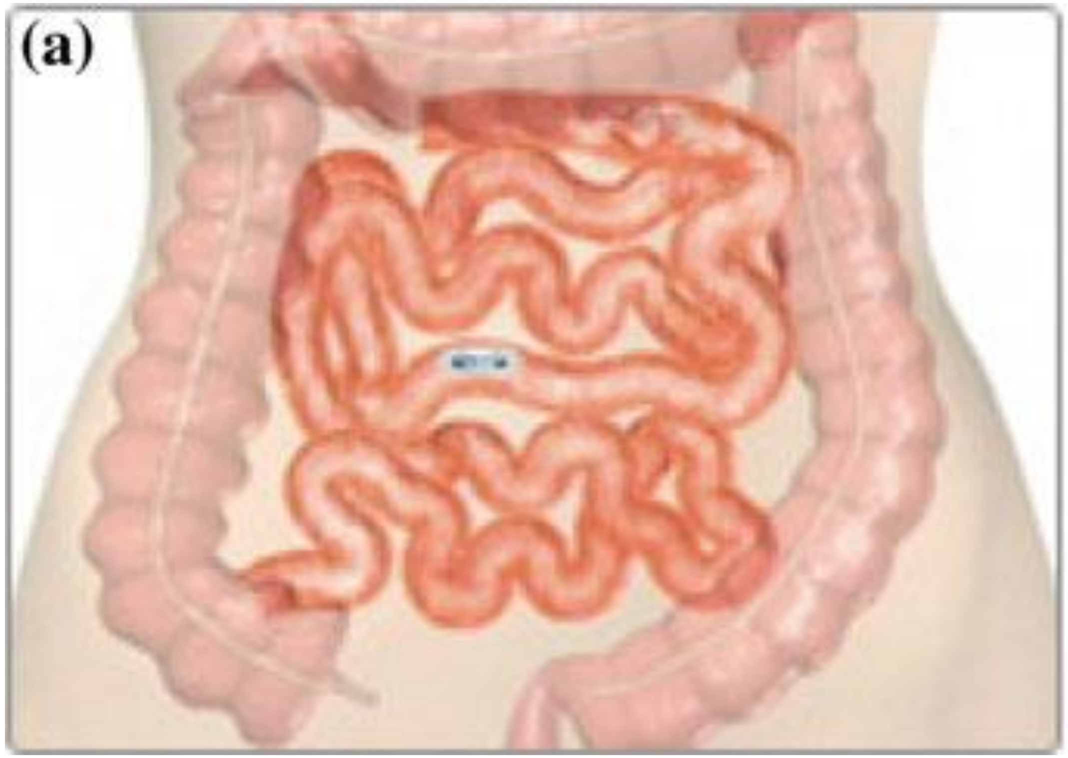


Figure: Wireless capsule endoscopy: (a) The patient swallows the capsule and capsule travels through the tubular intestinal path. The imaging is done through a circular lens and at periodic intervals. (b) Pillcam Capsule parts:
1 - Optical dome, 2 - Lens holder, 3 – Lens, 4 - Illuminating LEDs,
5 - CMOS imager, 6 – Battery, 7 - ASIC transmitter, 8 - Antenna

Automatic Image Segmentation for Video Capsule Endoscopy: V.B. Surya Prasath and R. Delhibabu

Summary

- The basic concept of CI was developed more than 45 years ago, but it took almost the last two decades for their potential to be recognized by a larger audience.
- Computational Intelligence-related techniques play an important role in state-of-the-art components and novel devices and services in science and engineering.
- The application of a **unimodal CI** technique will often be **insufficient** on its own to provide solutions to all the practical issues.
- **Hybrid systems** involving combinations of neural computation, fuzzy logic, and evolutionary algorithms, as well as traditional techniques, are a more **promising approach** for improving the performance of robot controllers.

Pop quiz

Note: Go to course page → Quizzes and answer the following questions.

Review Questions

1. Define computational intelligence.
2. Mention the different paradigms of computational intelligence.

Intelligent Agent

Lecture 2

Agent

- An agent is anything that can be viewed as **perceiving** its **environment** through **sensors** and **acting** upon that environment through actuators
- Can you suggest an example?
- **Human agent**
  - eyes, ears, and other organs for sensors – hands, legs, mouth, and other body parts for actuators
- **Robotic agent**
  - cameras and laser range finders for sensors – various motors for actuators

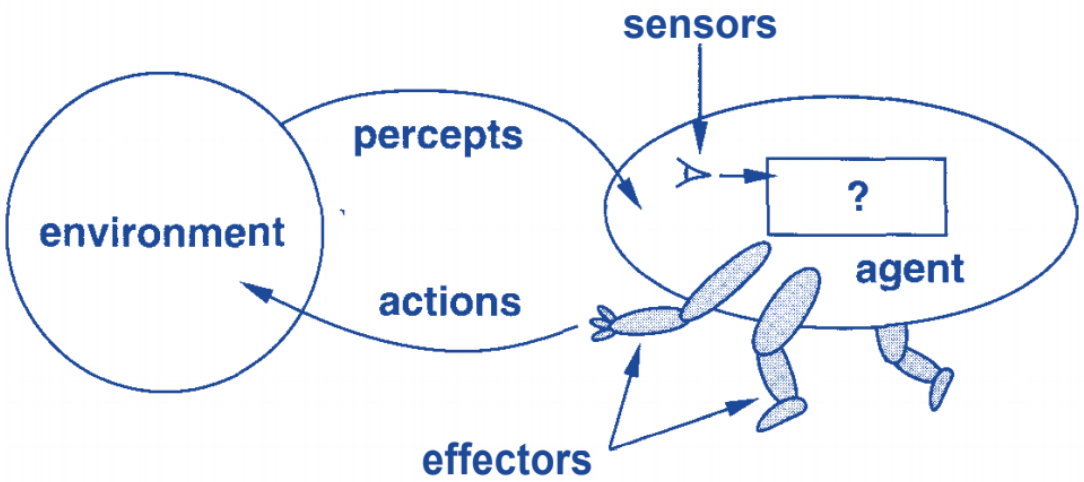
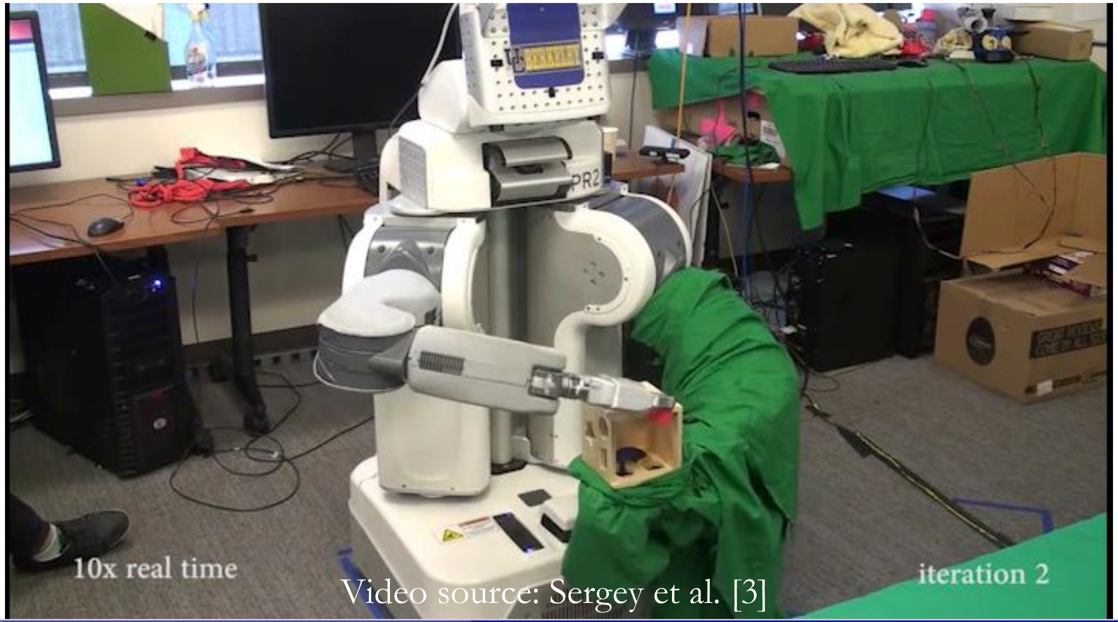


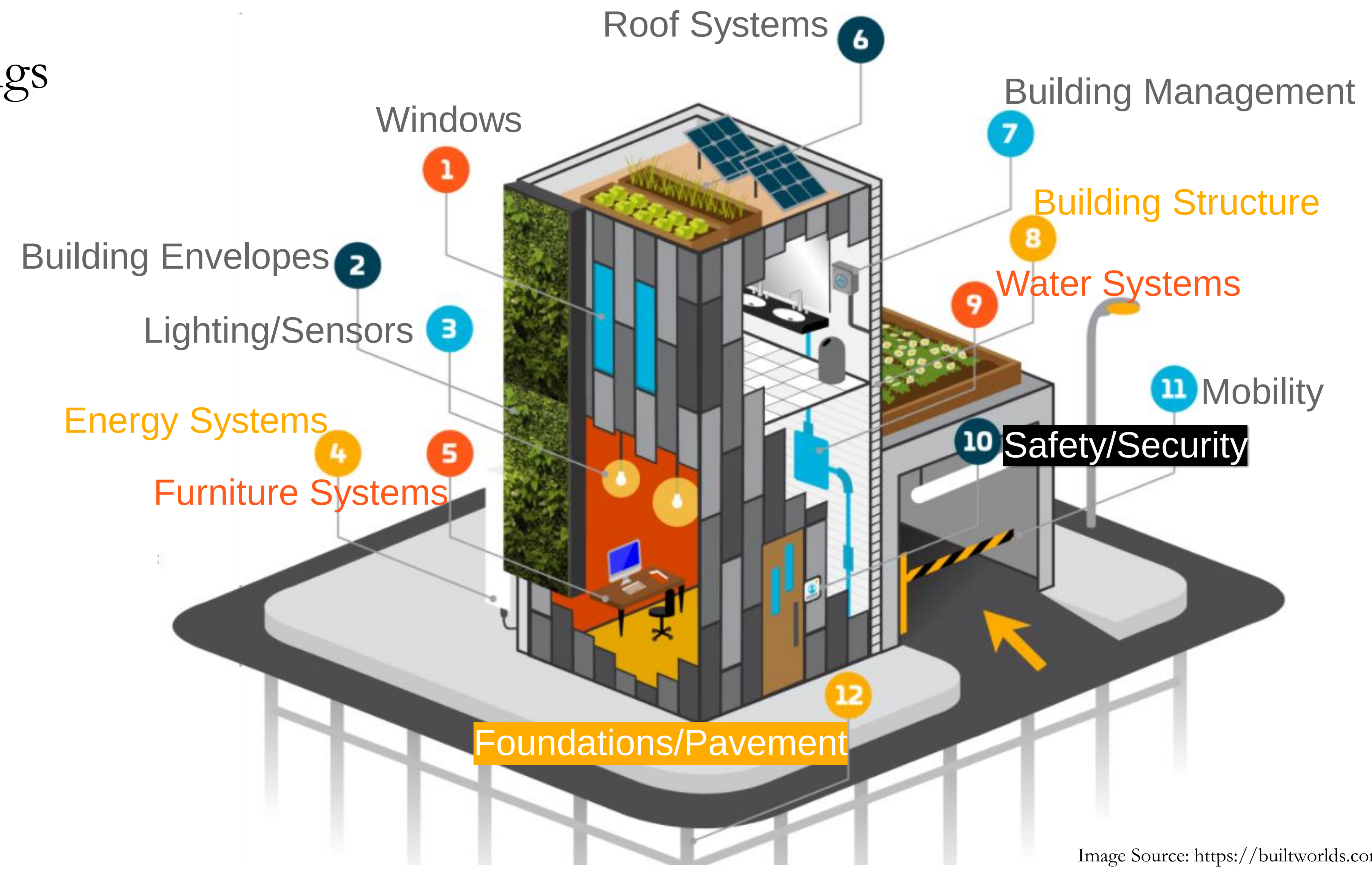
Figure: Agents interact with environments through sensors and effectors [2]



Video source: Sergey et al. [3]

Examples of Agents

- Intelligent buildings



course project

Image Source: https://builtworlds.com/

Examples of Agents

- Robots

Image Source: https://www.foxnews.com/

- Autonomous spacecraft

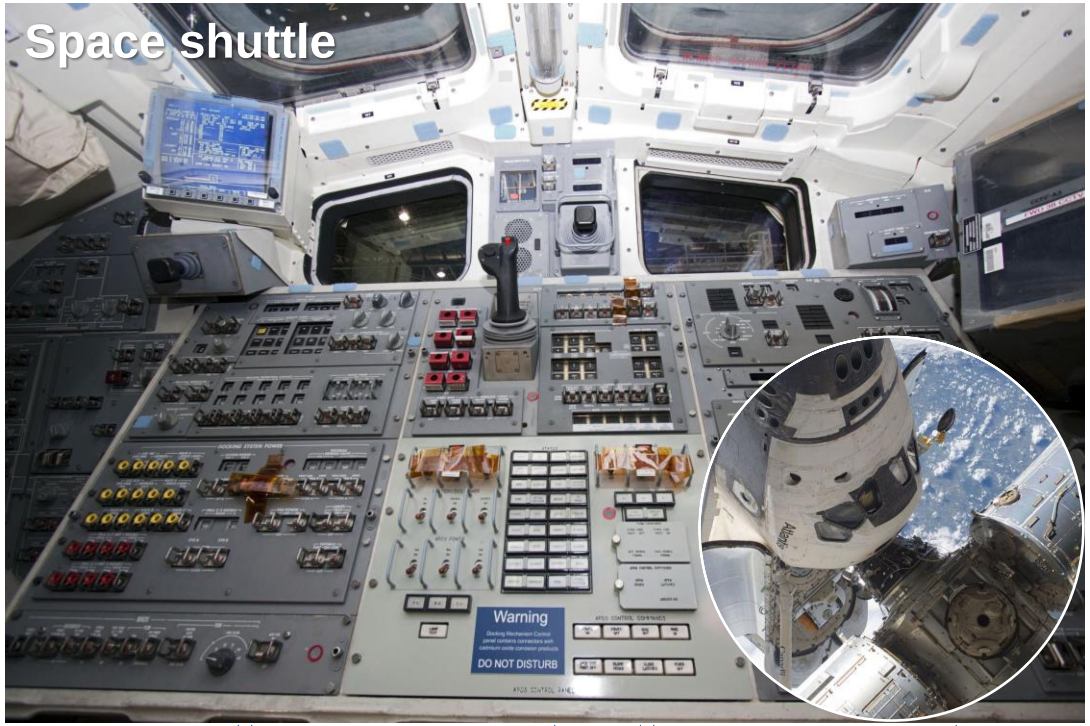


Image Sources: https://www.technologyreview.com/, https://www.electronicsweekly.com/, Wikipidia.com

Examples of Agents

- Web agents
  - Powerful techniques to **process information** intelligently and **offer** features based on **patterns** and **relationships** in the data.
  - Using the internet as a platform that not only gathers data at an ever-increasing pace but also systematically **transforms the raw data into actionable information**.
  - E.g.,
    - ✓ Netflix
    - ✓ Amazon
    - ✓ Google Ad Sense

Google AdSense

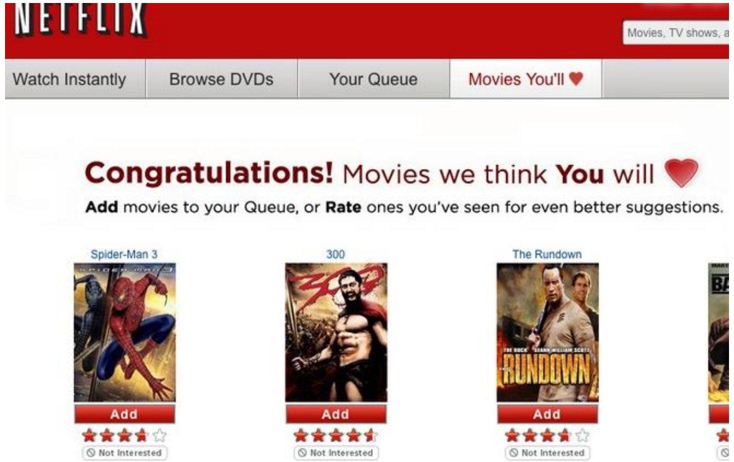


Image Sources: https://blog.watsi.org/, http://www.webdesignerdepot.com/

Intelligent Agent's Objective

- **Optimize energy** and **increase utility**
  - Example objectives
  - ✓Set a smart thermostat to turn on the A/C when building temps rise above 72°F
  - ✓Use motion sensors to trigger lights in less-trafficked hallways to save energy
  - ✓Install an access control system to improve building security and restrict access
  - ✓Manage connected facilities through a Building Information Modeling (BIM) system

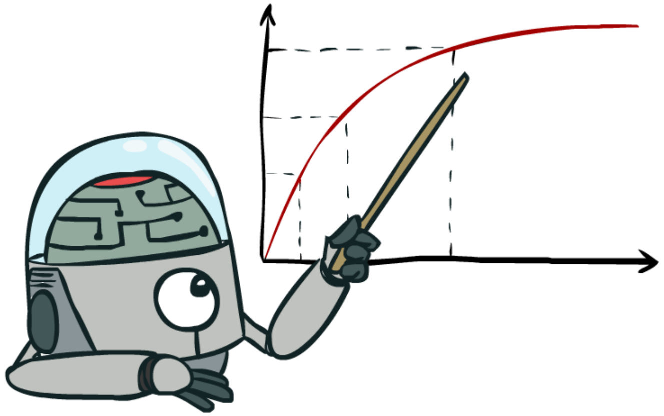


Image source: Anca Dragan's slides (CS 188: Introduction to Artificial Intelligence, Fall 2020 (berkeley.edu))

Problem Solving Agents

- Intelligent agents are supposed to act in such a way that the environment goes through a sequence of states that **maximizes the performance measure**.

- This specification is **difficult to translate** into a successful agent design.

- Let's simplify it by adopting a **goal and aim to satisfy it**.

Problem Solving Agents Cont.

- **Goals** help organize behavior by limiting objectives.
- A **goal** to be a set of world **states** – exactly those states in which the goal is satisfied.

- **Assumption:** environments that are known, observable, discrete and deterministic (i.e., each action has exactly one corresponding outcome).
  - action $A \rightarrow$ outcome R

- The process of **looking for a sequence of actions** that **reaches** the **goal** is called **search**.

- A search algorithm takes a **problem space as input and returns a solution** in the form of an **action sequence**.

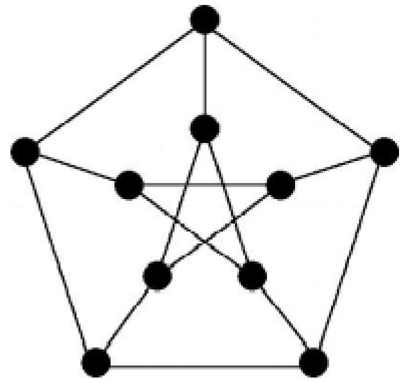
Formulating Problems and Solutions

- A problem can be defined formally by (5) components:

1. The initial state from which the agent starts
2. Operator **:** a description of possible actions available to the agent: ACTIONS(s)
3. A description of what each action does, i.e., the **transition model**, specified by a function $RESULT(s, a) = a'$.

State space:

- The **initial state, actions** and **transition model** implicitly defined the state space of the problem – the set of all states reachable from the initial state by any sequence of actions.
- It forms a **directed network** or graph in which the nodes are states and the edges between nodes are actions.
  - A **path** in the state space is a sequence of states connected by a sequence of actions.



Formulating Problems and Solutions Cont.

4. **Goal test:** it determines whether a given state is a **goal state**.
   - Frequently the goal test is intuitive (e.g., check if we arrived at the destination) – but note that it is also sometimes specified by an abstract property (e.g., "check mate").

5. **Path cost function:** It assigns a numeric cost to each path.
   - The agent chooses a cost function that reflects its own performance measure.
   - Commonly (but not always), the cost of a path is additive in terms of the individual actions along a path.
   - Denote the step cost to take action 'a' in state s, arriving in s' as: $c(s, a, s')$.