CECS545-Artificial Intelligence

Agents

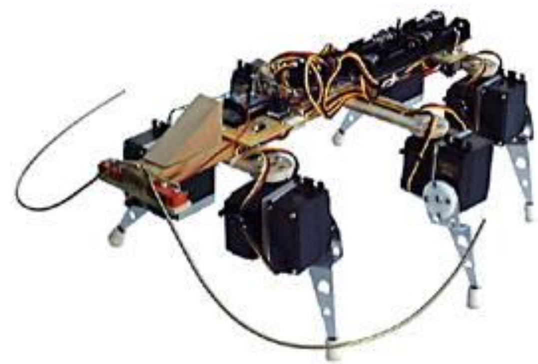
Dr. Roman Yampolskiy

What is an (Intelligent) Agent?

- Anything that can be *viewed as* **perceiving** its **environment** through **sensors** and **acting** upon that environment through its **effectors** to maximize progress towards its **goals**.

- This definition includes:
  - Robots, humans, programs

Examples of Agents



Humans	Programs	Robots
senses	keyboard, mouse, dataset	cameras, pads
body parts	monitor, speakers, files	motors, limbs

Rational Agents

A **rational agent** is one that does the right thing

- Need to be able to assess agent's performance
  - Should be independent of internal measures
- Ask yourself: has the agent acted rationally?
  - Not just dependent on how well it does at a task
- First consideration: evaluation of rationality

Autonomy in Agents

The **autonomy** of an agent is the extent to which its behaviour is determined by its own experience

- Extremes
  - No autonomy – ignores environment/data
  - Complete autonomy – must act randomly/no program
- Example: baby learning to crawl
- Ideal: design agents to have some autonomy
  - Possibly good to become more autonomous in time

What is an (Intelligent) Agent?

- **PAGE** (Percepts, Actions, Goals, Environment)

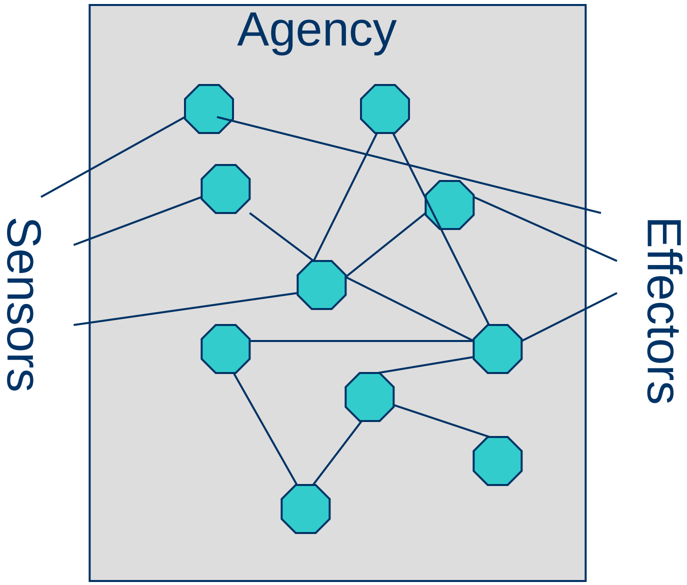
- Task-specific & specialized: well-defined goals and environment
- The notion of an agent is meant to be a tool for analyzing systems
  - It is not a different hardware or new programming languages

Intelligent Agents and AI

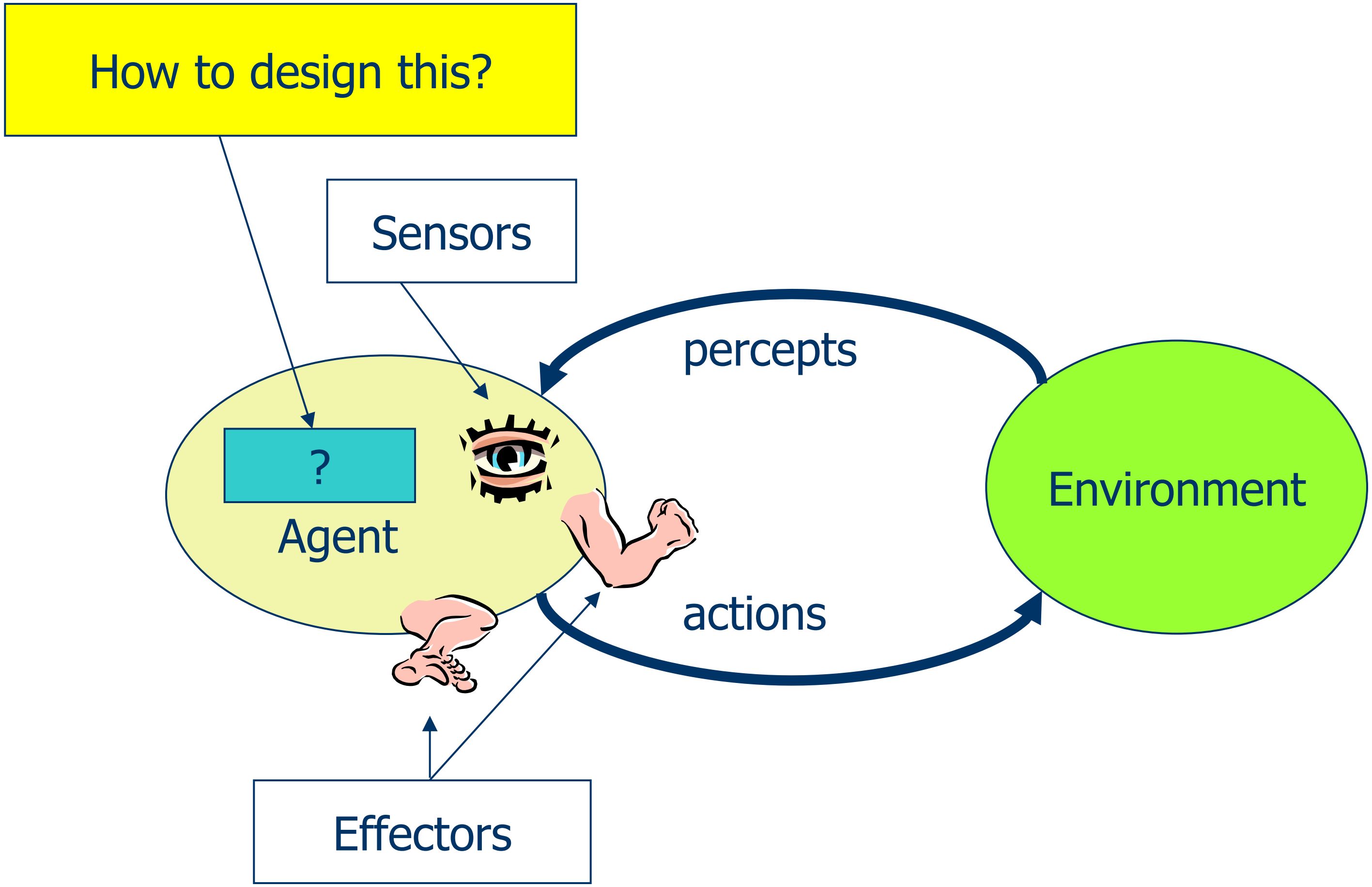
- **Example:** Human mind as network of thousands or millions of agents working in parallel. To produce real artificial intelligence, this school holds, we should build computer systems that also contain many agents and systems for arbitrating among the agents' competing results.

- Distributed decision-making and control

- Challenges:
  - Action selection: What next action to choose
  - Conflict resolution



Rational Agents



A Windshield Wiper Agent

How do we design a agent that can wipe the windshields when needed?

- Goals?
- Percepts?
- Sensors?
- Effectors?
- Actions?
- Environment?

A Windshield Wiper Agent (Cont'd)

- Goals: Keep windshields clean & maintain visibility
- Percepts: Raining, Dirty
- Sensors: Camera (moist sensor)
- Effectors: Wipers (left, right, back)
- Actions: Off, Slow, Medium, Fast
- Environment: Inner city, freeways, highways, weather

Towards Autonomous Vehicles

Interacting Agents

Collision Avoidance Agent (CAA)

- Goals: Avoid running into obstacles
- Percepts: Obstacle distance, velocity, trajectory
- Sensors: Vision, proximity sensing
- Effectors: Steering Wheel, Accelerator, Brakes, Horn, Headlights
- Actions: Steer, speed up, brake, blow horn, signal (headlights)
- Environment: Freeway

Lane Keeping Agent (LKA)

- Goals: Stay in current lane
- Percepts: Lane center, lane boundaries
- Sensors: Vision
- Effectors: Steering Wheel, Accelerator, Brakes
- Actions: Steer, speed up, brake
- Environment: Freeway

The Right Thing = The Rational Action

- **Rational Action:** The action that maximizes the expected value of the performance measure <u>given the percept sequence to date</u>

  – Rational = Best ?
  – Rational = Optimal ?
  – Rational = Omniscience ?
  – Rational = Clairvoyant ?
  – Rational = Successful ?

The Right Thing = The Rational Action

- **Rational Action:** The action that maximizes the expected value of the performance measure <u>given the percept sequence to date</u>

  - Rational = Best — Yes, to the best of its knowledge
  - Rational = Optimal — Yes, to the best of its abilities
  - Rational $\neq$ Omniscience
  - Rational $\neq$ Clairvoyant
  - Rational $\neq$ Successful

Behavior and performance of IAs

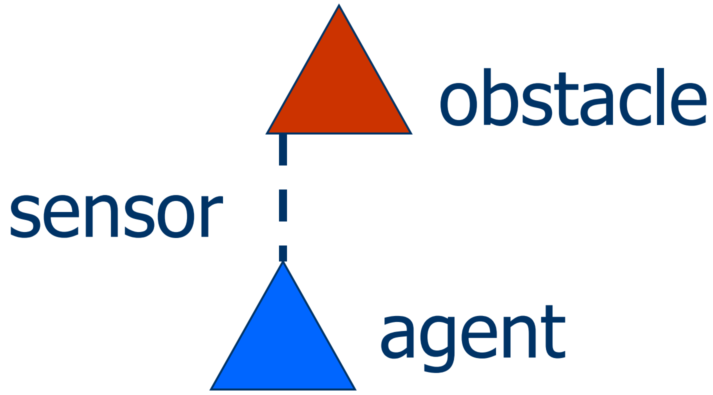
- **Perception** (sequence) to **Action Mapping:** $f : \mathcal{P}^* \to \mathcal{A}$
  - **Ideal mapping:** specifies which actions an agent ought to take at any point in time
  - **Example:** Look-Up-Table

- **Performance measure:** a *subjective* measure to characterize how successful an agent is (e.g., speed, power usage, accuracy, money, etc.)

- (degree of) **Autonomy:** to what extent is the agent able to make decisions and take actions on its own?

Look up table

Distance	Action
10	No action
5	Turn left 30 degrees
2	Stop



How is an Agent different from other software?

- Agents are **autonomous**, that is, they act on behalf of the user
- Agents contain some level of **intelligence**, from fixed rules to learning engines that allow them to adapt to changes in the environment
- Agents don't only act **reactively**, but sometimes also **proactively**

How is an Agent different from other software?

- Agents have **social ability**, that is, they communicate with the user, the system, and other agents as required
- Agents may also **cooperate** with other agents to carry out more complex tasks than they themselves can handle
- Agents may **migrate** from one system to another to access remote resources or even to meet other agents

Environment Types

- Characteristics
  - Accessible vs. inaccessible
  - Deterministic vs. nondeterministic
  - Episodic vs. nonepisodic
  - Hostile vs. friendly
  - Static vs. dynamic
  - Discrete vs. continuous

Environment Types

- Characteristics
  - Accessible vs. inaccessible
    - Sensors give access to **complete** state of the environment.
  - Deterministic vs. nondeterministic
    - The next state can be determined based on the current state and the action.
  - Episodic vs. nonepisodic (Sequential)
    - Episode: each perceive and action pairs
    - The quality of action does not depend on the previous episode.

Environment Types

- Characteristics
  - Hostile vs. friendly
  - Static vs. dynamic
    - Dynamic if the environment changes during deliberation
  - Discrete vs. continuous
    - Chess vs. driving

Environment types

Environment	Accessible	Deterministic	Episodic	Static	Discrete
Operating System	Yes	Yes	No	No	Yes
Virtual Reality	Yes	Yes	Yes/no	No	Yes/no
Office Environment	No	No	No	No	No
Mars	No	Semi	No	Semi	No

The environment types largely determine the agent design.

Structure of Intelligent Agents

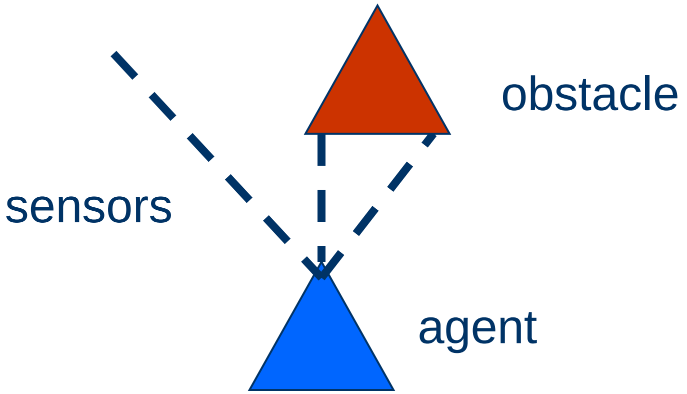
- **Agent program:** the implementation of $f : \mathcal{P}^* \rightarrow \mathcal{A}$, the agent's perception-action mapping

function Skeleton-Agent(*Percept*) **returns** *Action*
    memory $\leftarrow$ UpdateMemory(memory, *Percept*)
    Action $\leftarrow$ ChooseBestAction(memory)
    memory $\leftarrow$ UpdateMemory(memory, *Action*)
return *Action*

- **Architecture:** a device that can execute the agent program (e.g., general-purpose computer, specialized device, etc.)

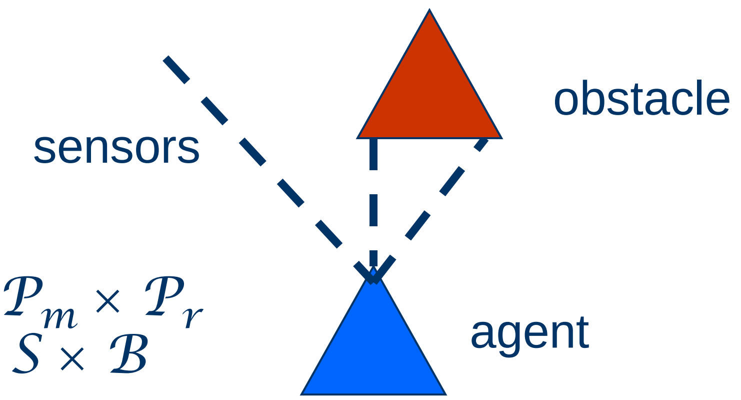
Using a look-up-table : $\mathcal{P}^* \rightarrow \mathcal{A}$

- **Example:** Collision Avoidance
  - Sensors: 3 proximity sensors
  - Effectors: Steering Wheel, Brakes
- How to generate?
- How large?
- How to select action?



Using a look-up-table: $\mathcal{P}^* \rightarrow \mathcal{A}$

- Example: Collision Avoidance
  - Sensors: 3 proximity sensors
  - Effectors: Steering Wheel, Brakes



- How to generate: for each $p \in \mathcal{P}_l \times \mathcal{P}_m \times \mathcal{P}_r$ generate an appropriate action, $a \in \mathcal{S} \times \mathcal{B}$

- How large: size of table = #possible percepts times # possible actions = $|\mathcal{P}_l|\ |\mathcal{P}_m|\ |\mathcal{P}_r|\ |\mathcal{S}|\ |\mathcal{B}|$
  E.g., P = {close, medium, far}^{3}
  A = {left, straight, right} $\times$ {on, off}
  then size of table = 27*3*2 = 162

- How to select action? Search.

Agent types

- Reflex agents
  - Reactive: No memory
- Reflex agents with internal states
  - W/o previous state, may not be able to make decision
    - E.g. brake lights at night.
- Goal-based agents
  - Goal information needed to make decision

Agent types

- Utility-based agents
  - How well can the goal be achieved (degree of happiness)
  - What to do if there are conflicting goals?
    - Speed and safety
  - Which goal should be selected if several can be achieved?

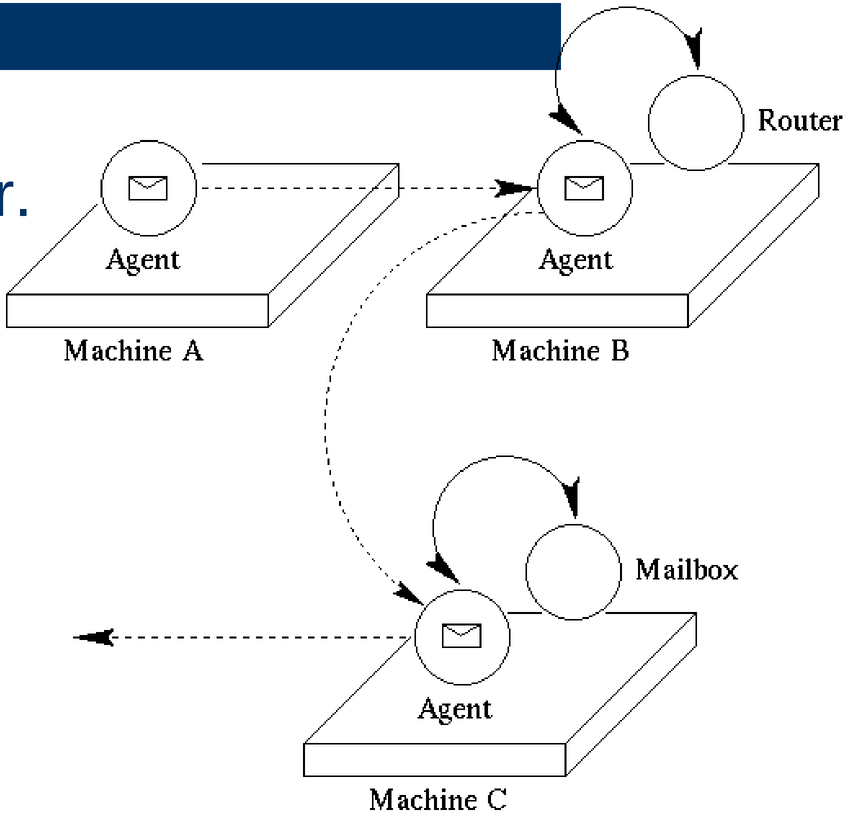
Reactive agents

- Reactive agents do not have internal symbolic models.
- Act by stimulus-response to the current state of the environment.
- Each reactive agent is simple and interacts with others in a basic way.
- Complex patterns of behavior emerge from their interaction.

- **Benefits:** robustness, fast response time
- **Challenges:** scalability, how intelligent?
  and how do you debug them?

Mobile agents

- Programs that can migrate from one machine to another.
- Execute in a platform-independent execution environment.
- Mobility not necessary or sufficient condition for agenthood.



A mail agent

Mobile agents

- Practical but non-functional advantages:
  - Reduced communication cost (e.g. from PDA)
  - Asynchronous computing (when you are not connected)

- Two types:
  - One-hop mobile agents (migrate to one other place)
  - Multi-hop mobile agents (roam the network from place to place)

Information agents

- Manage the explosive growth of information.
- Manipulate or collate information from many distributed sources.
- Information agents can be mobile or static.
- Examples:
  - BargainFinder comparison shops among Internet stores for CDs

Running Example

The RHINO Robot Museum Tour Guide

- Museum guide in Bonn
- Two tasks to perform
  - Guided tour around exhibits
  - Provide info on each exhibit

Internal Structure

- Second lot of considerations
  - Architecture and Program
  - Knowledge of the Environment
  - Reflexes
  - Goals
  - Utility Functions

Architecture and Program

- Program
  - Method of turning environmental input into actions
- Architecture
  - Hardware/software (OS etc.) on which agent's program runs

- RHINO's architecture:
  - Sensors (infrared, sonar, tactile, laser)
  - Processors (3 onboard, 3 more by wireless Ethernet)
- RHINO's program:
  - Low level: probabilistic reasoning, vision,
  - High level: problem solving, planning (first order logic)

Knowledge of Environment

- Knowledge of Environment (World)
  - Different to sensory information from environment
- World knowledge can be (pre)-programmed in
  - Can also be updated/inferred by sensory information
- Choice of actions informed by knowledge of...
  - Current state of the world
  - Previous states of the world
  - How its actions change the world
- Example: Chess agent
  - World knowledge is the board state (all the pieces)
  - Sensory information is the opponents move
  - Its moves also change the board state

RHINO’s Environment Knowledge

- Programmed knowledge
  - Layout of the Museum
    - Doors, exhibits, restricted areas
- Sensed knowledge
  - People and objects (chairs) moving
- Affect of actions on the World
  - Nothing moved by RHINO explicitly
  - But, people followed it around (moving people)

Reflexes

- Action on the world
  - In response only to a sensor input
  - Not in response to world knowledge
- Humans – flinching, blinking
- Chess – openings, endings
  - Lookup table (not a good idea in general)
    - 35^{100} entries required for the entire game
- RHINO: no reflexes?
  - Dangerous, because people get everywhere

Goals

- Always need to think hard about
  - What the goal of an agent is
- Does agent have internal knowledge about goal?
  - Obviously not the goal itself, but some properties
- Goal based agents
  - Uses knowledge about a goal to guide its actions
  - E.g., Search, planning
- RHINO
  - Goal: get from one exhibit to another
  - Knowledge about the goal: whereabouts it is
  - Need this to guide its actions (movements)

Utility Functions

- Knowledge of a goal may be difficult to pin down
  - For example, checkmate in chess
- But some agents have localised measures
  - Utility functions measure value of world states
  - Choose action which best improves utility (rational!)
  - In search, this is “Best First”
- RHINO: various utilities to guide search for route
  - Main one: distance from the target exhibit
  - Density of people along path

Details of the Environment

- Must take into account:
  - some qualities of the world
- Imagine:
  - A robot in the real world
  - A software agent dealing with web data streaming in
- Third lot of considerations:
  - Accessibility, Determinism
  - Episodes
  - Dynamic/Static, Discrete/Continuous

Accessibility of Environment

- Is everything an agent requires to choose its actions available to it via its sensors?
  - If so, the environment is fully accessible
- If not, parts of the environment are inaccessible
  - Agent must make informed guesses about world
- RHINO:
  - “Invisible” objects which couldn’t be sensed
  - Including glass cases and bars at particular heights
  - Software adapted to take this into account

Determinism in the Environment

- Does the change in world state
  - Depend only on current state and agent's action?
- Non-deterministic environments
  - Have aspects beyond the control of the agent
  - Utility functions have to guess at changes in world
- Robot in a maze: deterministic
  - Whatever it does, the maze remains the same
- RHINO: non-deterministic
  - People moved chairs to block its path

Episodic Environments

- Is the choice of current action
  - Dependent on previous actions?
  - If not, then the environment is episodic
- In non-episodic environments:
  - Agent has to plan ahead:
    - Current choice will affect future actions
- RHINO:
  - Short term goal is episodic
    - Getting to an exhibit does not depend on how it got to current one
  - Long term goal is non-episodic
    - Tour guide, so cannot return to an exhibit on a tour

Static or Dynamic Environments

- Static environments don't change
  - While the agent is deliberating over what to do
- Dynamic environments do change
  - So agent should/could consult the world when choosing actions
  - Alternatively: anticipate the change during deliberation
  - Alternatively: make decision very fast
- RHINO:
  - Fast decision making (planning route)
  - But people are very quick on their feet

Discrete or Continuous Environments

- Nature of sensor readings / choices of action
  - Sweep through a range of values (continuous)
  - Limited to a distinct, clearly defined set (discrete)
- Chess: discrete
- RHINO: continuous
  - Visual data can be considered continuous
  - Choice of actions (directions) also continuous

RHINO's Solution to Environmental Problems

- Museum environment:
  - Inaccessible, non-episodic, non-deterministic, dynamic, continuous
- RHINO constantly updates plan as it moves
  - Solves these problems very well
  - Necessary design given the environment

The End!