Agents and Environments

Stephen G. Ware

CSCI 4525 / 5525

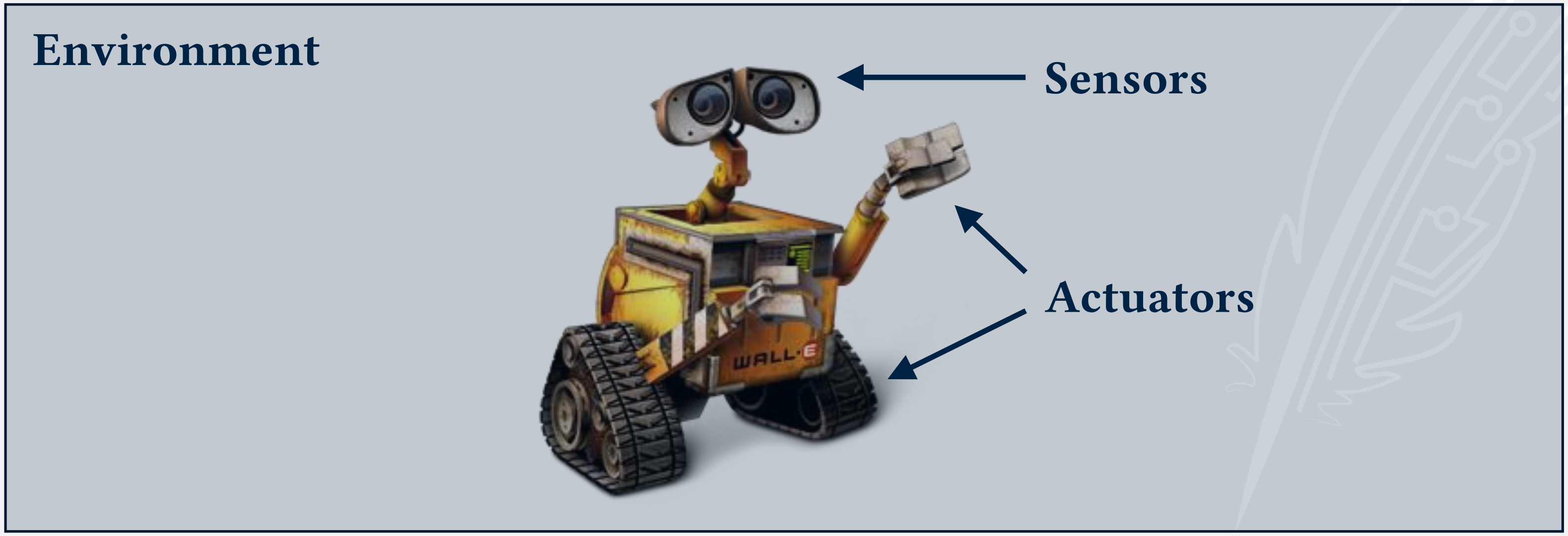
Narrative Intelligence LAB

THE UNIVERSITY *of* NEW ORLEANS

Agents

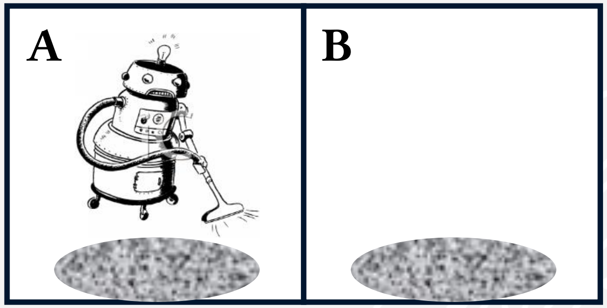
An **agent** (software or hardware) has:

- **Sensors** that perceive its environment
- **Actuators** that change its environment



Vacuum World

- Environment:
  - Room A
  - Room B
  - Dirt
- Vacuum Agent:
  - Room Sensor
  - Dirt Sensor
  - Motor
  - Vacuum



Agent Function

An **agent function** maps any given percept sequence to an action.

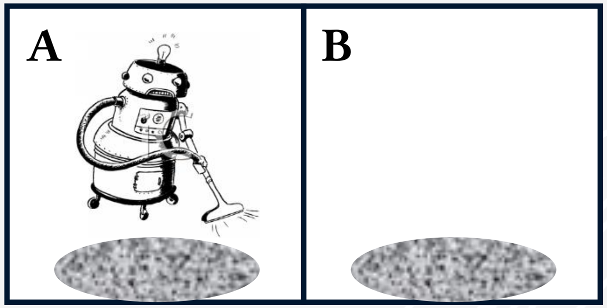
Agent functions are generally implemented in software as **agent programs**.

Vacuum Agent Function

Percept sequence:

1. You are in Room A.
2. Room A is dirty.

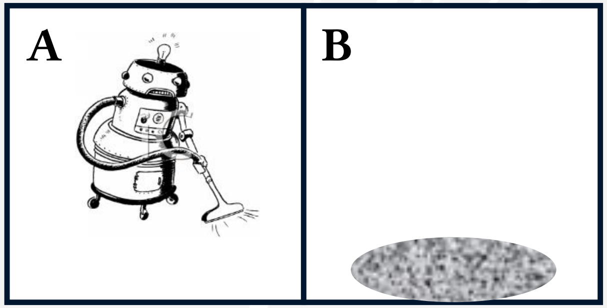
Action: suck



Percept sequence:

1. You are in Room A.
2. Room A is clean.

Action: go right

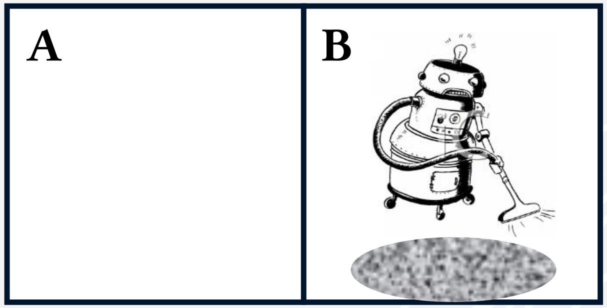


Vacuum Agent Function

Percept sequence:

1. You are in Room A.
2. Room A is clean.
3. You are in Room B.
4. Room B is dirty.

Action: suck



NIL

Performance Measures

An agent's success is defined according to some **performance measure**, which must be defined for each task the agent is expected to perform.

Vacuum Performance

Example:

Score 1 point for every pile of dirt sucked up in an 8 hour shift.

Agent sucks dirt, then dumps it out, then sucks it up again, then dumps it out again, etc…

“Excuse me, do you know what time it is?”

“Yes, I do.”

Vacuum Performance

Better:

Score 1 point for each clean room every minute.

Lesson: Design performance measures according to what you want in the environment, not how you think the agent should behave.

Rational Agent

Things to consider:

- The performance measure
- The agent's prior knowledge
- The percept sequence to date
- The agent's available actions

Rational Agent

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.

Omniscience

- Rational agents maximize *expected* performance, not necessarily *actual* performance.
- Agents are not expected to be omniscient.

Coping with Limited Knowledge

- Some agents know what they do not know and act to improve their knowledge via **information gathering**, such as sensing and exploration.
- Some agents can update their own agent programs based on observations, called **learning**.
- Agents which gather information and learn are **autonomous**, whereas agent which rely only on prior knowledge lack autonomy and are fragile.

A Bug's Life

Dung Beetle

Sphex Wasp

NIL

Environments

- Single Agent vs. Multi-Agent
- Episodic vs. Sequential
- Static vs. Dynamic
- Fully Observable vs. Partially Observable
- Deterministic vs. Stochastic
- Known vs. Unknown
- Discrete vs. Continuous

Single Agent vs. Multi-Agent

- **Single Agent**: Everything else in the environment can be treated as an object, rather than an agent.
- **Multi-Agent**: The actions of other objects in the environment will depend on the agent's actions, and should therefore be viewed as other agents.
  - **Competitive**: Maximizing my performance may hurt another's performance.
  - **Cooperative**: Through coordinated action, we can both achieve better performance.

Episodic vs. Sequential

- **Episodic**: An agent's current choice of actions does not depend on previous choices.
- **Sequential**: An agent's current choice of actions will effect which actions are available in the future, and thus some kind of planning is required.

Static vs. Dynamic

- **Static**: The agents are the only cause of change in the environment. Time is not a factor.
- **Dynamic**: The environment may change independently of the agents. Time is a factor.
- **Semidynamic**: The agents are the only source of change, but time is a factor.

Fully vs. Partially Observable

- **Fully Observable**: Every relevant piece of information is available on demand. The agent does not need to track the environment's state.
- **Partially Observable**: Some relevant information is missing and must be assumed or gathered.
  - Information is unavailable due to the agent's state.
  - Appropriate sensors are missing or malfunctioning.
  - Sensory data is noisy.

Deterministic vs. Uncertain

- **Deterministic**: Given the current state, we can know exactly the outcome of an action.
- **Uncertain**: The next state cannot be known with full confidence because the environment is not fully observable or action outcomes unpredictable.
  - **Stochastic**: Outcome probabilities are known.
  - **Nondeterministic**: Probabilities not known.

Known vs. Unknown

- **Known**: The rules (even if they are stochastic) of the environment are known in advance.
- **Unknown**: The rules of the environment are not known and must be learned.

Discrete vs. Continuous

- **Discrete**: The environment's state and time and the agent's actions can be easily divided into clearly labeled classes.
- **Continuous**: State and/or time and/or actions cannot be clearly divided into distinct classes.

Environments

- Single Agent vs. Multi-Agent
- Episodic vs. Sequential
- Static vs. Dynamic
- Fully Observable vs. Partially Observable
- Deterministic vs. Stochastic
- Known vs. Unknown
- Discrete vs. Continuous

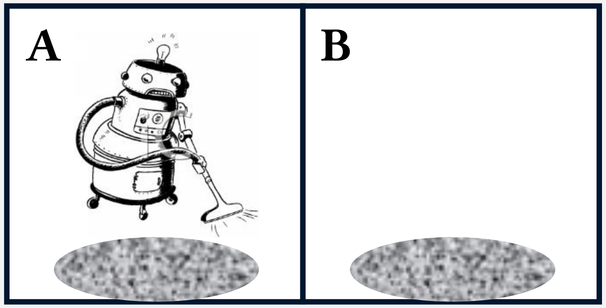
The Easiest Environment

- **Single Agent** vs. Multi-Agent
- **Episodic** vs. Sequential
- **Static** vs. Dynamic
- **Fully Observable** vs. Partially Observable
- **Deterministic** vs. Stochastic
- **Known** vs. Unknown
- **Discrete** vs. Continuous

The Real World

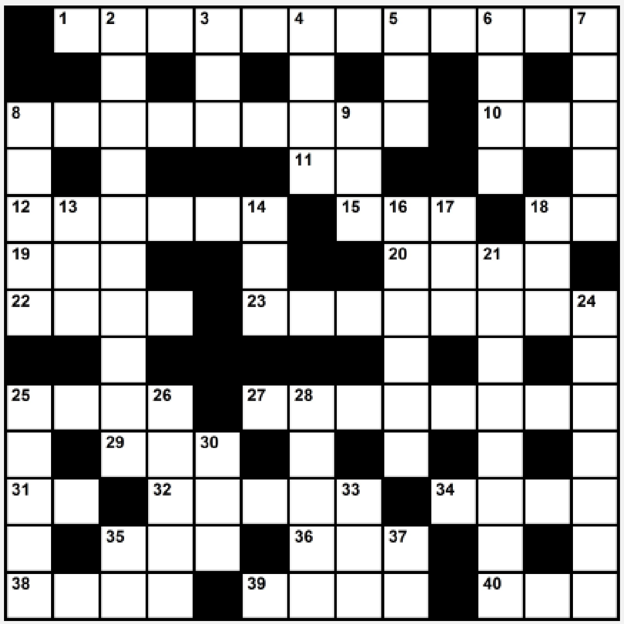
- Single Agent vs. **Multi-Agent**
- Episodic vs. **Sequential**
- Static vs. **Dynamic**
- Fully Observable vs. **Partially Observable**
- Deterministic vs. **Stochastic**
- Known vs. **Unknown**
- Discrete vs. **Continuous**

Vacuum World



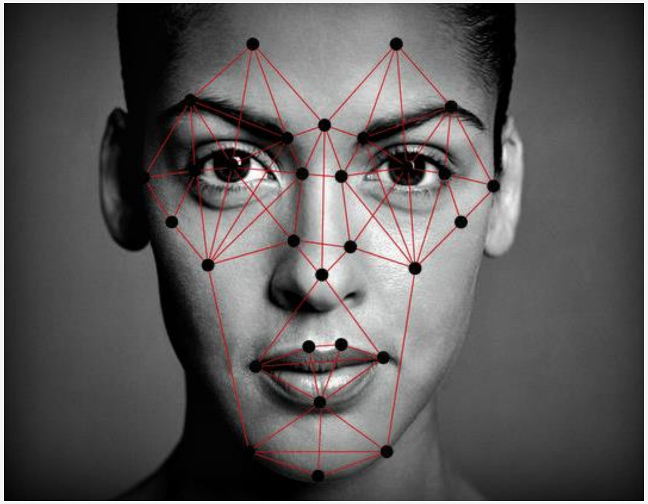
Observable:	Partially
Agents:	Single
Deterministic:	Deterministic
Episodic:	Episodic
Static:	Static
Discrete:	Discrete

Crossword Puzzle



Observable:	Fully
Agents:	Single
Deterministic:	Deterministic
Episodic:	Sequential
Static:	Static
Discrete:	Discrete

Facial Recognition



Observable:	Fully
Agents:	Single
Deterministic:	Deterministic
Episodic:	Episodic
Static:	Static
Discrete:	Continuous

Chess

Observable:	Fully
Agents:	Multi
Deterministic:	Deterministic
Episodic:	Sequential
Static:	Static
Discrete:	Discrete

Chess with a Clock

Observable:	Fully
Agents:	Multi
Deterministic:	Deterministic
Episodic:	Sequential
Static:	Semidynamic
Discrete:	Discrete

Poker

Observable:	Partially
Agents:	Multi
Deterministic:	Stochastic
Episodic:	Sequential
Static:	Static
Discrete:	Discrete

Intelligent Tutoring System

Observable:	Partially
Agents:	Multi
Deterministic:	Stochastic
Episodic:	Sequential
Static:	Dynamic
Discrete:	Discrete

Driving a Taxi

Observable:	Partially
Agents:	Multi
Deterministic:	Nondeterministic
Episodic:	Sequential
Static:	Dynamic
Discrete:	Continuous

Developing AI Solutions

Solving real world problems in AI:

1. Make as many simplifying assumptions as possible.
2. Solve the easy problem.
3. Relax one of the simplifying assumptions.
4. Solve the harder problem.
5. Repeat steps 3 and 4 until you have solved the real world problem.