CS486/686: Introduction to Artificial Intelligence
Lecture 2 - Agents and Abstraction

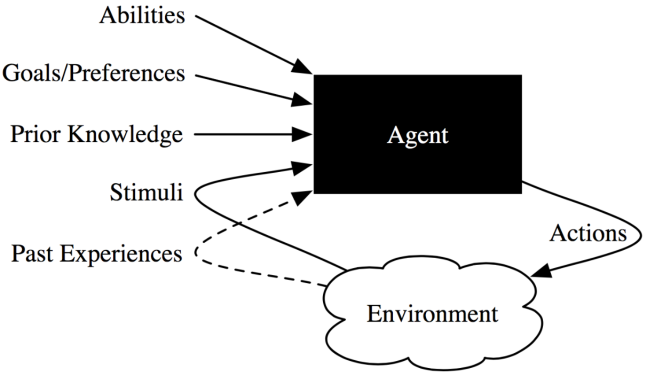
Jesse Hoey & Victor Zhong

School of Computer Science, University of Waterloo

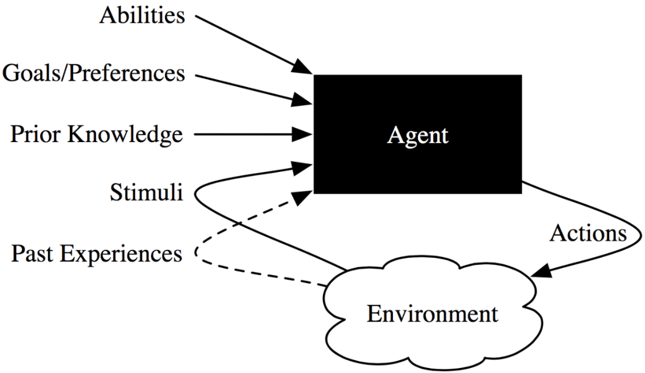
January 6, 2025

Readings: Poole & Mackworth 1.1

Situated Agent



Situated Agent



- An **agent** is an entity that performs actions in its **environment**
- Agent+Environment=**world**
- Inside black box: **belief state**

Four Example Application Domains (From Book)

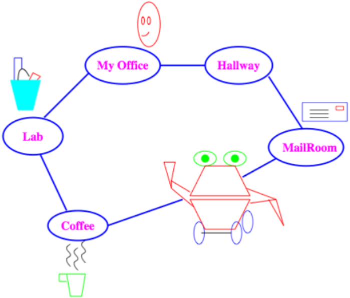
- **Autonomous delivery robot** roams around an office environment and delivers coffee, parcels,...
- **Diagnostic assistant** helps a human troubleshoot problems and suggests repairs or treatments, e.g., electrical problems and medical diagnosis
- **Intelligent tutoring system** teaches students in some subject area
- **Trading agent** buys goods and services on your behalf

Four Example Application Domains (From Book)

- **Autonomous delivery robot** roams around an office environment and delivers coffee, parcels,...
- **Diagnostic assistant** helps a human troubleshoot problems and suggests repairs or treatments, e.g., electrical problems and medical diagnosis
- **Intelligent tutoring system** teaches students in some subject area
- **Trading agent** buys goods and services on your behalf

Let's talk about the **autonomous delivery robot**

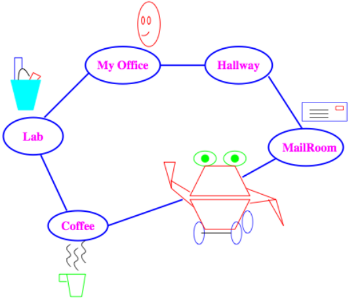
Domain for Delivery Robot

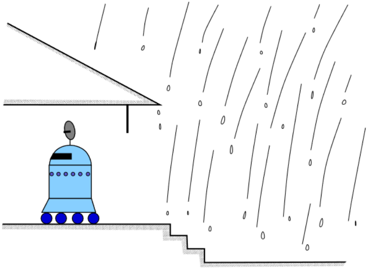


The robot must:

- Deliver coffee & mail when needed

Domain for Delivery Robot





The robot must:

- Deliver coffee & mail when needed
- Avoid getting wet

Autonomous Delivery Robot

- **Abilities:** movement, speech, pickup and place objects, sense weather

Autonomous Delivery Robot

- **Abilities:** movement, speech, pickup and place objects, sense weather
- **Stimuli:** informatiom about its environment from cameras, sonar, sound, laser range finders, or keyboards

Autonomous Delivery Robot

- **Abilities:** movement, speech, pickup and place objects, sense weather
- **Stimuli:** informatiom about its environment from cameras, sonar, sound, laser range finders, or keyboards
- **Prior knowledge:** its capabilities, objects it may encounter, maps

Autonomous Delivery Robot

- **Abilities:** movement, speech, pickup and place objects, sense weather
- **Stimuli:** informatiom about its environment from cameras, sonar, sound, laser range finders, or keyboards
- **Prior knowledge:** its capabilities, objects it may encounter, maps
- **Past experience:** which actions are useful and when, what objects are there, how its actions affect its position

Autonomous Delivery Robot

- **Abilities:** movement, speech, pickup and place objects, sense weather
- **Stimuli:** informatiom about its environment from cameras, sonar, sound, laser range finders, or keyboards
- **Prior knowledge:** its capabilities, objects it may encounter, maps
- **Past experience:** which actions are useful and when, what objects are there, how its actions affect its position
- **Goals:** what it needs to deliver and when, tradeoffs between acting quickly and acting safely, effects of getting wet

What Does the Delivery Robot Need to Do?

- Determine where user is, where coffee is, ...

What Does the Delivery Robot Need to Do?

- Determine where user is, where coffee is, ...
- Find a path between locations

What Does the Delivery Robot Need to Do?

- Determine where user is, where coffee is, ...
- Find a path between locations
- Plan how to carry out multiple tasks

What Does the Delivery Robot Need to Do?

- Determine where user is, where coffee is, ...
- Find a path between locations
- Plan how to carry out multiple tasks
- Make default assumptions about where user is

What Does the Delivery Robot Need to Do?

- Determine where user is, where coffee is, ...
- Find a path between locations
- Plan how to carry out multiple tasks
- Make default assumptions about where user is
- Make tradeoffs under uncertainty: should it go near the stairs or outside?

What Does the Delivery Robot Need to Do?

- Determine where user is, where coffee is, ...
- Find a path between locations
- Plan how to carry out multiple tasks
- Make default assumptions about where user is
- Make tradeoffs under uncertainty: should it go near the stairs or outside?
- Learn from experience

What Does the Delivery Robot Need to Do?

- Determine where user is, where coffee is, ...
- Find a path between locations
- Plan how to carry out multiple tasks
- Make default assumptions about where user is
- Make tradeoffs under uncertainty: should it go near the stairs or outside?
- Learn from experience
- Sense and act in the world, avoid obstacles, pickup and put down coffee, deliver mail

Knowledge Representation

- **Knowledge**: information used to solve tasks
- **Representation**: data structures used to encode knowledge
- **Knowledge base (KB)**: representation of all knowledge
- **Model**: relationship of KB to world
- **Level of Abstraction**: How accurate is the model

Knowledge Representation

- **Knowledge**: information used to solve tasks
- **Representation**: data structures used to encode knowledge
- **Knowledge base (KB)**: representation of all knowledge
- **Model**: relationship of KB to world
- **Level of Abstraction**: How accurate is the model
- Non-AI:
  - Specify how to compute something
  - Specify what the next step is
  - **Programmer** figures out how to do the computation

Knowledge Representation

- **Knowledge**: information used to solve tasks
- **Representation**: data structures used to encode knowledge
- **Knowledge base (KB)**: representation of all knowledge
- **Model**: relationship of KB to world
- **Level of Abstraction**: How accurate is the model
- Non-AI:
  - Specify how to compute something
  - Specify what the next step is
  - **Programmer** figures out how to do the computation
- AI:
  - Specify what needs to be computed
  - Specify how the world works
  - **Agent** figures out how to do the computation

Dimensions of Complexity

- The textbook defines 9 “dimensions of complexity” that make up the agent design space
- This is not a complete set (i.e., designing an agent is not as simple as selecting an option for each dimension)
- Can be multiple values in a dimension: values go from simple to complex
- Much of the history of AI can be seen as starting from the simple and adding in complexity in some of these dimensions

Dimensions of Complexity

1. **Modularity**: flat → modular → hierarchical
2. **Planning horizon**: non-planning → finite horizon → indefinite horizon → infinite horizon
3. **Representation**: explicit states → features → individuals and relations
4. **Computational limits**: perfect rationality → bounded rationality
5. **Learning**: knowledge is given → knowledge is learned
6. **Uncertainty**: fully observable → partially observable
   world dynamics: deterministic → stochastic
7. **Preference**: goals → complex preferences
8. **Reasoning by number of agents**: single agent → adversarial → multiagent
9. **Interactivity**: offline → online

Dimension 2: Planning Horizon

How far the agent looks into the future when deciding what to do

- **Static:** world does not change

Dimension 2: Planning Horizon

How far the agent looks into the future when deciding what to do

- **Static:** world does not change
- **Finite horizon:** agent reasons about a fixed finite number of time steps

Dimension 2: Planning Horizon

How far the agent looks into the future when deciding what to do

- **Static:** world does not change
- **Finite horizon:** agent reasons about a fixed finite number of time steps
- **Indefinite horizon:** agent is reasoning about finite, but not predetermined, number of time steps (e.g. until goal completion)

Dimension 2: Planning Horizon

How far the agent looks into the future when deciding what to do

- **Static:** world does not change
- **Finite horizon:** agent reasons about a fixed finite number of time steps
- **Indefinite horizon:** agent is reasoning about finite, but not predetermined, number of time steps (e.g. until goal completion)
- **Infinite horizon:** the agent plans for going on forever (process oriented)

Dimension 3: Representation

Much of modern AI is about finding compact representations and exploiting that compactness for computational gains

Dimension 3: Representation

Much of modern AI is about finding compact representations and exploiting that compactness for computational gains

An agent can reason in terms of:

- **Explicit states**—a state directly represents one way the world could be

Dimension 3: Representation

Much of modern AI is about finding compact representations and exploiting that compactness for computational gains

An agent can reason in terms of:

- **Explicit states**—a state directly represents one way the world could be
- **Features** or **propositions**
  - It is often more natural to describe states in terms of features
  - 30 binary features can represent $2^{30} = 1,073,741,824$ states

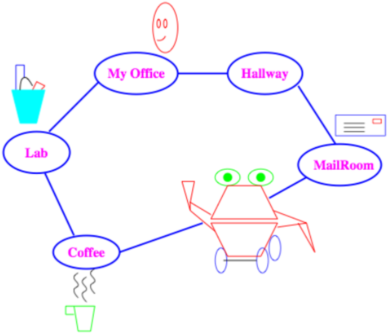
Dimension 3: Representation

Much of modern AI is about finding compact representations and exploiting that compactness for computational gains

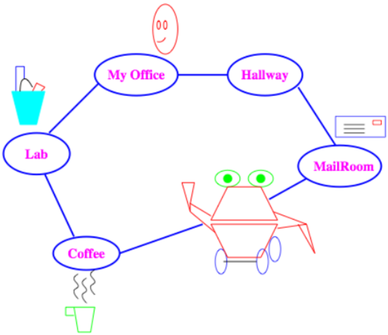
An agent can reason in terms of:

- **Explicit states**—a state directly represents one way the world could be
- **Features** or **propositions**
  - It is often more natural to describe states in terms of features
  - 30 binary features can represent $2^{30} = 1,073,741,824$ states
- **Individuals** and **relations**
  - There is a feature for each relationship on each tuple of individuals
  - Often we can reason without knowing the individuals or when there are infinitely many individuals

Example: Delivery Robot

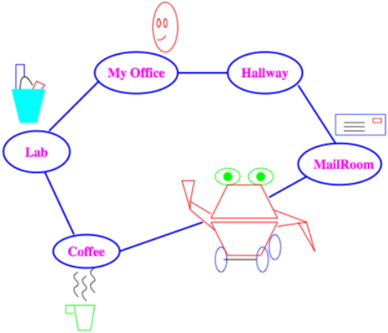


Example: Delivery Robot



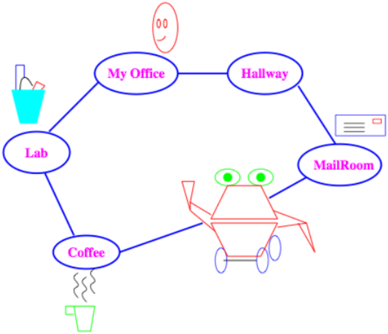
- Explicit: enumeration of all worlds: $s_1, s_2, s_3, \ldots$

Example: Delivery Robot



- Explicit: enumeration of all worlds: $s_1, s_2, s_3, \ldots$
- Features: robot location, user location, robot has coffee?, ...

Example: Delivery Robot



- Explicit: enumeration of all worlds: $s_1, s_2, s_3, \ldots$
- Features: robot location, user location, robot has coffee?, ...
- Relations: robot moves (clockwise + or counter-clockwise -)
  $\forall m \in \{+, -\}, \ell \in \{1, 2, 3...\}, move(m) : \ell' \leftarrow (\ell \; m \; 1)\%5$

Dimension 4: Computational Limits

Do we have time to calculate exact solutions?

Dimension 4: Computational Limits

Do we have time to calculate exact solutions?

- **Perfect rationality:**
  Agent always chooses the optimal (i.e. highest utility) action

Dimension 4: Computational Limits

Do we have time to calculate exact solutions?

- **Perfect rationality:**
  Agent always chooses the optimal (i.e. highest utility) action
- **Bounded rationality:**
  Agent chooses a possibly sub-optimal action given its limited computational capacity (e.g. chess bots)
  - **Satisficing** solution (good enough)
  - **Approximately optimal** solution (how far off?)

Dimension 5: Learning from Experience

Whether the model is fully specified a priori:

- Knowledge is **given**
- Knowledge is **learned from data or past experience**

Dimension 6: Uncertainty

What the agent can determine the state from the observations:

- **Fully-observable**: the agent knows the state of the world from the observations
- **Partially-observable**: there can be many states that are possible given an observation

Uncertain World Dynamics

If the agent knew the initial state and the action, could it predict the resulting state?

Uncertain World Dynamics

If the agent knew the initial state and the action, could it predict the resulting state?

The dynamics can be:

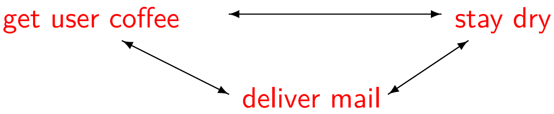
- **Deterministic**: the state resulting from carrying out an action in state is determined from the action and the state
- **Stochastic**: there is uncertainty over the states resulting from executing a given action in a given state

Dimension 7: Goals or Complex Preferences

- **Achievement goal** is a goal to achieve; can be a complex logical formula
- **Maintenance goal** is a goal to be maintained
- **Complex preferences** that may involve tradeoffs between various desiderata, perhaps at different times; can be either ordinal or cardinal (e.g., utility)
- Examples: coffee delivery robot, medical doctor

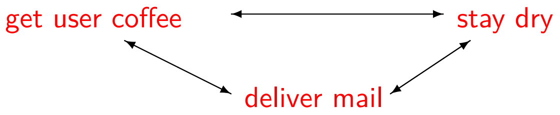
Example: Complex Preferences

Delivery Robot



Example: Complex Preferences

Delivery Robot



- Goals may conflict

  e.g. can't deliver mail and coffee at the same time
- Goals may be combinatorial

  e.g. user may not want coffee if he doesn't get mail
- Goals may change

  e.g. - when wet, robot can't deliver mail

  - user switches from coffee to kale juice

Dimension 8: Reasoning by Number of Agents

- **Single agent** reasoning is where an agent assumes that any other agents are part of the environment (delivery robot)
- **Adversarial** reasoning considers another agent that acts in opposition to our goals (AlphaGo)
- **Multiagent** reasoning is when an agent needs to reason strategically about the reasoning of other agents (robot soccer, trading agents)

Dimension 8: Reasoning by Number of Agents

- **Single agent** reasoning is where an agent assumes that any other agents are part of the environment (delivery robot)
- **Adversarial** reasoning considers another agent that acts in opposition to our goals (AlphaGo)
- **Multiagent** reasoning is when an agent needs to reason strategically about the reasoning of other agents (robot soccer, trading agents)

Agents can have their own goals: cooperative, competitive, or goals can be independent of each other

Next

- Uninformed Search (Poole & Mackworth chapter 3)