Lecture 2 - Agents and Abstraction

Jesse Hoey
School of Computer Science
University of Waterloo

September 17, 2015

Readings: Poole & Mackworth 1.3-1.10
Situated Agent

Abilities
Goals/Preferences
Prior Knowledge
Observations
Past Experiences

Agent

Environment

Actions
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, medical diagnosis. 
- **Intelligent tutoring system** teaches students in some subject area. 
- **Trading agent** buys goods and services on your behalf.
Domain for Delivery Robot
Example inputs:

- **Abilities:** movement, speech, pickup and place objects, sense weather
- **Observations:** 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 Jesse’s office is. Where coffee is...
- Find a path between locations.
- Plan how to carry out multiple tasks.
- Make default assumptions about where Jesse is.
- Make tradeoffs under uncertainty: should it go near the stairs or outside?
- Learn from experience.
- Sense the world, avoid obstacles, pickup and put down coffee.
Assistance for Persons with Dementia: COACH

University of Toronto/University of Waterloo
Example: COACH Handwashing Assistant

movie from slides here:
/Users/jhoey/project/talks/icvs/coachanimationOct2011.mp4
COACH (Handwashing System)

- **Abilities:** Provide passive assistive prompting
- **Observations:** Video images, audio feeds
- **Prior knowledge:** Effects of dementia, steps of handwashing, affect/culture
- **Past experience:** Prior behaviours of user, effects of prompts
- **Goals:** Get hands clean, maintain independence, reduce caregiver burden
What does the COACH need to do?

- Sense the world, determine what steps the user has completed
- Estimate the user’s current responsiveness, awareness and emotion
- Provide timely prompts only if needed
- Call for assistance if it can’t help
- Learn from experience.
Automated Scheduling Assistant
Automated Scheduling Assistant

- **Abilities:** Schedule patients for hospital procedures
- **Observations:** Arrival times, resource levels,
- **Prior knowledge:** Common illnesses, resource consumption,
- **Past experience:** Patient health trajectories, human scheduling and prioritization
- **Goals:** Maximize health, minimize resource utilization, minimize doctor fatigue
What does the Scheduling Assistant need to do?

- Maintain count of how many people are in the system and where they are
- Predict future usage of resources
- Analyze patient current and future health states
- Admit patients
- Request further resources
Intelligent Tutoring System

- **Abilities:** Present information, give tests
- **Observations:** test results, facial expressions, questions, what the student is concentrating on
- **Prior knowledge:** subject material, primitive strategies, student knowledge
- **Past experience:** common errors, effects of teaching strategies
- **Goals:** the students should master subject material, gain social skills, study skills, inquisitiveness, interest
Common Tasks of the Domains

- **Modeling the environment** Build models of the physical environment, patient, or information environment.

- **Evidential reasoning or perception** Given observations, determine what the world is like.

- **Action** Given a model of the world and a goal, determine what should be done.

- **Learning from past experiences** Learn about the specific case and the population of cases.
Knowledge Representation

problem → solve → solution

represent → compute → output

informal → formal

represent interpret
Knowledge Representation

A good representation should be
- Rich enough to express the problem
- Close to the problem: compact, natural and maintainable
- Amenable to efficient computation
- Amenable to elicitation from people, data and experiences

Very similar to software engineering.
A **symbol** is a meaningful physical pattern that can be manipulated.

A **symbol system** creates, copies, modifies and destroys symbols.

**physical symbol system hypothesis** (Newell & Simon, 1976):

A physical symbol system has the necessary and sufficient means for general intelligent action.

implies that: AI on a computer is possible in theory, but not necessarily feasible in practice
Knowledge can be represented at different levels of abstraction.

- Level chosen depends on the task at hand.
- Low-level: more detail, more precision, more difficult computation.
- High-level: less detail, less precision, easier computation.
- Often multiple levels of abstraction are necessary: **hierarchy**.
Two strategies for building agents

movie from slides here: *chaplin* — *mt.mp4*

simplify/constrain the environment simplify the agents
Reasoning and acting

Reasoning is the computation required to determine what an agent should do.

- **Design time reasoning and computation** is carried out by the designer of the agent.
- **Offline computation** is the computation done by the agent before it has to act.
- Background knowledge and data $\leadsto$ **knowledge base** .
- **Online computation** is the computation that’s done by an agent between receiving information and acting.
Solutions

Given a problem, what is a solution?

- **Optimal** according to some measure of quality e.g. utility
- **Satisficing**: meets all necessary criteria
- **Approximately optimal**: e.g. within 10% of optimal
- **Probable**: is likely to be right (but could be wrong...really wrong)
Good decisions can have bad outcomes. Bad decisions can have good outcomes.

Information can be valuable because it leads to better decisions: value of information.

You have to trade off computation time and solution quality: an anytime algorithm can provide a solution at any time; given more time it can produce better solutions.

You don’t only need to be concerned about finding the right answer, but about acquiring the appropriate information, and computing it in a timely manner.
Research proceeds by making simplifying assumptions, and gradually reducing them.

Each simplifying assumption gives a dimension of complexity
  - Can be multiple values in a dimension: values go from simple to complex
    - Simplifying assumptions can be relaxed in various combinations

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

- Flat $\rightarrow$ modular $\rightarrow$ hierarchical
- Explicit states $\rightarrow$ features $\rightarrow$ objects and relations
- Static $\rightarrow$ finite stage $\rightarrow$ indefinite stage $\rightarrow$ infinite stage
- Fully observable $\rightarrow$ partially observable
- Deterministic $\rightarrow$ stochastic dynamics
- Goals $\rightarrow$ complex preferences
- Single-agent $\rightarrow$ multiple agents
- Knowledge is given $\rightarrow$ knowledge is learned from experience
- Perfect rationality $\rightarrow$ bounded rationality
Modularity

- Model at one level of abstraction: **flat**
- Model with interacting modules that can be understood separately: **modular**
- Model with modules that themselves are (recursively) decomposed into modules: **hierarchical**
- Flat representations are adequate for simple systems, but complex biological systems, computer systems, organizations are all hierarchical
- A flat description is either continuous or discrete. Hierarchical reasoning is often a hybrid of continuous and discrete.
Much of modern AI is about finding compact representations and exploiting that compactness for computational gains. A agent can reason in terms of:

- **explicit states** — a state is one way the world could be
Much of modern AI is about finding compact representations and exploiting that compactness for computational gains. A agent can reason in terms of:

- **explicit states** — a state is one way the world could be
- **features** or **propositions**.
  - It’s often more natural to describe states in terms of features.
  - 30 binary features can represent $2^{30} = 1,073,741,824$ states.
Much of modern AI is about finding compact representations and exploiting that compactness for computational gains. A agent can reason in terms of:

- **explicit states** — a state is one way the world could be
- **features** or **propositions**.
  - It’s 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: Handwashing Task: Plansteps

- **Dirty**: a → b
- **Dirty, Dry**: a → e
- **Soapy**: e → d
- **Soapy, Dry**: d
- **Water On**: d → g
- **Clean, Wet**: g → h
- **Clean, Dry**: h → j
- **Water Off**: j → k
- **Behavior: Rinse**: d

...how far the agent looks into the future when deciding what to do.

- **Static:** world does not change
- **Finite stage:** agent reasons about a fixed finite number of time steps
- **Indefinite stage:** agent is reasoning about finite, but not predetermined, number of time steps
- **Infinite stage:** the agent plans for going on forever (process oriented)
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.
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.
Example: Sensing and Dynamics Uncertainty
Example: COACH

University of Toronto/University of Waterloo
Goals or complex preferences

- **achievement goal** is a goal to achieve. This 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. Either ordinal or cardinal (e.g., utility)

**Examples:** coffee delivery robot, medical doctor
Example: Complex Preferences

handwashing assistant:

get hands clean → user independence

caregiver burden →

Single agent or multiple agents

- **Single agent** reasoning is where an agent assumes that any other agents are part of the environment. (delivery robot)
- **Multiple agent** 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
Learning from experience

Whether the model is fully specified a priori:

- knowledge is given
- knowledge is learned from data or past experience
Perfect rationality or bounded rationality

- **Perfect rationality:** the agent can determine the best course of action, without taking into account its limited computational resources.

- **Bounded rationality:** the agent must make good decisions based on its perceptual, computational and memory limitations.
Dimensions of Complexity

- Flat or modular or hierarchical
- Explicit states or features or objects and relations
- Static or finite stage or indefinite stage or infinite stage
- Fully observable or partially observable
- Deterministic or stochastic dynamics
- Goals or complex preferences
- Single-agent or multiple agents
- Knowledge is given or knowledge is learned from experience
- Perfect rationality or bounded rationality
Delivery Robot

- **flat** → modular → hierarchical
- **explicit states** → features → objects and relations
- static → finite stage → **indefinite stage** → infinite stage
- fully observable → **partially observable**
- deterministic → **stochastic** actions
- **goals** → complex preferences
- **single agent** → multiple agents
- **knowledge is given** → learned
- **perfect rationality** → bounded rationality
Handwashing problem

- flat $\rightarrow$ modular $\rightarrow$ hierarchical
- explicit states $\rightarrow$ features $\rightarrow$ objects and relations
- static $\rightarrow$ finite stage $\rightarrow$ indefinite stage $\rightarrow$ infinite stage
- fully observable $\rightarrow$ partially observable
- deterministic $\rightarrow$ stochastic actions
- goals $\rightarrow$ complex preferences
- single agent $\rightarrow$ multiple agents
- knowledge is given $\rightarrow$ learned
- perfect rationality $\rightarrow$ bounded rationality
Next:

- Read Poole & Mackworth chapter 2.1-2.3
- Uninformed Search (Poole & Mackworth chapter 3)
- Informed Search (Poole & Mackworth chapter 4)