LECTURE 4: PRACTICAL REASONING

http://www.csc.liv.ac.uk/~mjw/pubs/imas/

An Introduction to Multiagent Systems
Practical reasoning is reasoning directed towards actions. The relevant considerations are provided by what the agent desires/values/cares about and what the agent believes.

Distinguish practical reasoning from theoretical reasoning.

Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the process of figuring out what to do: 1 Practical Reasoning

Theoretical reasoning is directed towards beliefs. (Bratman)
Human practical reasoning consists of two activities:

- deliberation
- means-ends reasoning

The outputs of deliberation are intentions.

Deciding how to achieve these states of affairs.

Deciding what state of affairs we want to achieve.
1. Intentions pose problems for agents, who need to determine ways of achieving them.

2. Intentions provide a "filter" for adopting other intentions, which resources to devoting about.

3. Agents track the success of their intentions, and are inclined to try again if their attempts fail.

If an agent's first attempt to achieve \( \phi \) fails, then all other things try again if their attempts fail.

If an agent's first attempt to achieve \( \phi \) fails, then all other things try again if their attempts fail.

If I have an intention to \( \phi \), you would expect me to adopt an intention such that \( \phi \) and \( \phi \) are mutually exclusive.

If I have an intention to \( \phi \), you would expect me to adopt an intention such that \( \phi \) and \( \phi \) are mutually exclusive.

\( \phi \) being equal, if \( \phi \) will try an alternative plan to achieve \( \phi \).

An Introduction to Multiagent Systems
4. Agents believe their intentions are possible. That is, they believe there is at least some way that the intentions could be brought about.

5. Agents do not believe they will not bring about their intentions. It would not be rational of me to adopt an intention to $\phi$ if I believed $\phi$ was not possible.

6. Under certain circumstances, agents believe they will bring about their intentions. It would not normally be rational of me to believe that I would bring my intentions about; intentions can fail. Moreover, it does not make sense that if I believe $\phi$ is inevitable that I would adopt it as an intention.
Lecture 4
An Introduction to Multiagent Systems

7. Agents need not intend all the expected side effects of their intentions. If I believe $\phi$ and I intend that $\phi$, I do not necessarily intend $\phi$. Also, (intentions are not closed under implication.) If I believe $\phi$ and I intend that $\phi$, I do not necessarily intend $\phi$. Also, (intentions are not closed under implication.)

This last problem is known as the *side effect or packaged deal* problem. I may believe that going to the dentist involves pain, and I may also intend to go to the dentist — but this does not imply that I intend to suffer pain!
Notice that intentions are much stronger than mere desires.
Since the early 1970s, the field of planning has been well-developed. Planning algorithms have been proposed, and the theory of planning has long been assumed that some form of AI planning system will be a central component of any artificial agent. Building largely on the early work of Fikes & Nilsson, many algorithms have been proposed, and the theory of planning is essentially automatable. Progarmming: the design of a course of action that will achieve some desired goal. Within the symbolic AI community, it has long been assumed that the design of a course of action that will achieve some desired goal. Planning is closely concerned with the design of artificial agents.
What is Means-Ends Reasoning?

Basic idea is to give an agent:

- representation of a goal to achieve the goal.
- representation of the environment;
- representation of actions it can perform; and
- representation of goal/intention to achieve;

Essentially, this is

autonomous programming.

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
plan to achieve goal

Planner

possible actions
environment
state of
goal

possible actions
environment
state of
goal

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
Question: How do we represent...

- goal to be achieved;
- state of environment;
- actions available to agent;
- plan itself.
We'll illustrate the techniques with reference to the blocks world.

- Contains a robot arm, 2 blocks (A and B) of equal size, and a table-top.
To represent this environment, need an ontology.
Here is a representation of the blocks world described above:

<table>
<thead>
<tr>
<th></th>
<th>OnTable</th>
<th>Clear</th>
<th>On</th>
<th>OnTable</th>
<th>Clear</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td>C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Use the closed world assumption: anything not stated is assumed to be false.
A goal is represented as a set of formulae.

\{ \text{OnTable}(A), \text{OnTable}(B), \text{OnTable}(C) \}

Here is a goal:

A goal is represented as a set of formulae.
Actions are represented using a technique that was developed in the STRIPS planner.

Actions

Each action has:

- a name
- a precondition list
- a delete list
- an add list

Each of these may contain variables.

List of facts made true by executing the action.

List of facts that are no longer true after action is performed.

List of facts which must be true for action to be executed.

List which may have arguments.

Each of these may contain variables.
Example 1:
The stack action occurs when the robot arm places the object \( x \) it is holding is placed on top of object \( y \).

\[
\begin{align*}
\text{Stack}(x, y) & \quad \text{pre} \quad \text{Clear}(y) \land \text{Holding}(x) \\
\text{del} \quad \text{Clear}(y) \land \text{Holding}(x) \\
\text{add} \quad \text{ArmEmpty} \land \text{On}(x, y)
\end{align*}
\]
Stack and Unstack are inverses of one-another.

\[
\begin{align*}
\text{add } \& \text{del } & \text{Holding}(x) \lor \text{Clear}(x) \\
\text{pre } & \text{On}(x,y) \lor \text{ArmEmpty}(x,y) \\
\text{pre } & \text{Clear}(x) \lor \text{ArmEmpty}(x,y) \\
\text{Unstack}(x,y) &
\end{align*}
\]

Example 2: The unstack action occurs when the robot arm picks an object up from on top of another object.
Example 3:
The action occurs when the arm picks up an object from the table.

\[
\begin{align*}
\text{add } & \text{Holding}(x) \\
\text{del } & \text{OnTable}(x) \lor \text{ArmEmpty} \\
\text{pre } & \text{Clear}(x) \lor \text{OnTable}(x) \lor \text{ArmEmpty} \\
\text{Pickup}(x) &
\end{align*}
\]
Example 4:
The putdown action occurs when the arm places the object $x$ onto the table.

\[ \text{pre} \text{PutDown}(x) \land \text{del} \text{Holding}(x) \land \text{add} \text{ArmEmpty} \]

The putdown action occurs when the arm places the object $x$. 

- Example 4:
A sequence (list) of actions, with variables replaced by constants.

What is a plan?
A first pass at an implementation of a practical reasoning agent:

Agent Control Loop Version 1

1. while true
2. observe the world;
3. update internal world model;
4. deliberate about what intention to achieve next;
5. use means-ends reasoning to get a plan for the intention;
6. execute the plan;
7. end while

(We will not be concerned with stages (2) or (3).)

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
Problem: deliberation and means-ends reasoning processes are not instantaneous.

Suppose the agent starts deliberating at time \( t_1 \) and begins executing the plan at time \( t_2 \). Time to deliberate is

\[
t_1 - t_2 = \text{time to deliberate}
\]

and time for means-ends reasoning is

\[
t_0 - t_1 = \text{time for means-ends reasoning}
\]

They have a time cost.

Problem: deliberation and means-ends reasoning processes are not instantaneous.
Deliberation is only half of the problem: the agent still has to determine how to achieve the intention.

Further suppose that deliberation is optimal in that if it selects some intention to achieve, then this is the best thing for the agent. (Maximises expected utility.)

So at time \( t_0 \), the agent has selected an intention to achieve that intention. It would have been optimal if it had been achieved at \( t_0 \). But unless deliberation is vanishingly small, then the agent runs the risk that the intention selected is no longer optimal by the time the agent has fixed upon it.

Deliberation is calculative rationality.

This is calculative rationality.

An Introduction to Multiagent Systems
So, this agent will have overall optimal behavior in the following circumstances:

1. When deliberation and means-ends reasoning take a vanishingly small amount of time; or
2. When the world is guaranteed to remain static while the agent is deliberating and performing means-ends reasoning, so that the assumptions upon which the choice of intention to achieve and plan to achieve the intention remain valid until the time at which the world is observed is guaranteed to remain optimal at time $t^0$; or
3. When an intention that is optimal when achieved at time $t^0$ remains optimal until time $t^2$ (the time at which the agent has found a course of action to achieve the intention).

So, this agent will have overall optimal behavior in the following circumstances:
Let's make the algorithm more formal.

\begin{verbatim}
8. end while
7. execute
6. \textbf{while} true do
5. get next percept \textbf{do}
4. plan := \emptyset
3. deliberate := I
2. \textbf{while} true do
1. B := 0
   \textit{initialize beliefs}
   Agent Control Loop Version 2
\end{verbatim}

Let's make the algorithm more formal.
How does an agent deliberate?

1. Begin by trying to understand what the options available to you are.
2. Choose between them, and commit to some of them.

Chosen options are then intentions.
The deliberate function can be decomposed into two distinct functional components: option generation and filtering. In order to select between competing alternatives, an agent uses a function, options, which takes the agent's current beliefs and intentions, and from them determines a set of possible alternatives (desires). In which the agent chooses between competing alternatives, the agent uses a filter function. Represent option generation via a function, options, which in which the agent generates a set of possible alternatives; filtering from them determines a set of options (= desires).
Agent Control Loop Version 3
1.
2. \( B := B_0; \)
3. \( I := I_0; \)
4. while true do
5. \( \) get next percept \( \rho; \)
6. \( B := brf(B, \rho); \)
7. \( D := options(B, I); \)
8. \( I := filter(B, D, I); \)
9. \( \pi := plan(B, I); \)
10. \( execute(\pi) \)
11. end while
Commitment Strategies


Some time in the not-so-distant future, you are having trouble with your new household robot. You say, “Willie, bring me a beer.” The robot replies, “OK boss. The robot repl...
The following commitment strategies are commonly discussed in the literature of rational agents:

- **Blind commitment**: An agent will continue to maintain an intention as long as it believes that it has been achieved.
- **Single-minded commitment**: An agent will continue to maintain an intention until it believes that either the intention has been achieved or it is no longer possible to achieve the intention.
- **Open-minded commitment**: An agent will continue to maintain an intention as long as it is still believed possible.
- **Fanatical commitment**: An agent will continue to maintain an intention as long as it is still believed possible.
- **Blindly committed**: Blindly committed agents will continue to believe the intention has been achieved until it is no longer believed.

The following commitment strategies are commonly discussed in the literature of multiagent systems:
An agent has commitment both to ends (i.e., the state of affairs) and means (i.e., the mechanism via which the agent wishes to achieve the state of affairs). Currently, our agent control loop is overcommitted, both to means and ends.

Modification: replan if ever a plan goes wrong.
Agent Control Loop Version 4

1. $B := B_0$;

2. $I := I_0$;

3. while true do

4. get next percept $\rho$;

5. $B := \text{brf}(B, \rho)$;

6. $D := \text{options}(B, I)$;

7. $I := \text{filter}(B, D, I)$;

8. $\pi := \text{plan}(B, I)$;

9. while not empty($\pi$) do

10. $\alpha := \text{hd}(\pi)$;

11. execute($\alpha$);

12. $\pi := \text{tail}(\pi)$;

13. end-while

14. get next percept $\rho$;

15. $B := \text{brf}(B, \rho)$;

16. if not sound($\pi, I, B$) then

17. $\pi := \text{plan}(B, I)$;

18. end-if

19. end-while

20. end-while
Still overcommitted to intentions: Never stops to consider

Modification: stop to determine whether intentions have
succeeded or whether they are impossible:

(Stoical-minded commitment:

•

whether or not its intentions are appropriate.

An Introduction to Multiagent Systems
AgentControlLoopVersion5

2. while True do
3.    get next percept
4.    if not sound then
5.        plan =: v
6.        filter =: g
7.        options =: D
8.        thepercept =: I
9.        (f, q) := g
10.       end-if
11.      end-while
12.     execute
13.     tail
14.     get next percept
15.     (u) := w
16.     options =: D
17.     filter =: g
18.     plan =: v
19.     end-while
20. end-while

An Introduction to Multiagent Systems
Our agent gets to reconsider its intentions once every time it has completely executed a plan to achieve its current intentions; or it believes its current intentions are no longer possible; or it believes it has achieved its current intentions; or it has completely executed a plan to achieve its current intentions around the outer control loop; i.e., when:

- It has finished executing every action.

This is limited in the way that it permits an agent to reconsider its intentions.

Modification: Reconsider intentions after executing every action.
Lecture 4

Agent Control Loop Version 6

1. while true do
2. get next percept
3. filter
4. plan
5. if not sound
6. options
7. filter
8. options
9. get next percept
10. if not impossible or succeeded then
11. plan
12. execute
13. execute
14. get next percept
15. get next percept
16. options
17. filter
18. options
19. plan
20. end-if
21. end-while
22. end-while

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
But intention reconsideration is costly!

A dilemma:

But intention reconsideration is costly!

Solution: Incorporate an explicit meta-level control component.

That decides whether or not to reconsider.

- An agent that consistently reconsider its intentions may spend insufficient time actually working to achieve them, and hence runs the risk of never actually achieving them.

- An agent that does not stop to reconsider its intentions even after it is clear that they cannot be achieved, or that there is no longer any reason for achieving them, will continue attempting to achieve them; sufficiently often will continue attempting to achieve its intentions sufficiently often will continue attempting to achieve them; and hence insufficienent time actually working to achieve them, and hence.

http://www.csc.liv.ac.uk/~mjw/pubs/tmas/
Agent Control Loop

1. while true do
2. get next percept
3. if reconsider(df) then
4. options := D
5. get next percept
6. if sound then
7. filter := D
8. options := D
9. end-if
10. while not (empty or succeeded) do
11. if reconsider(df) then
12. options := D
13. filter := D
14. end-if
15. end-while
16. end-while

http://www.csc.liv.ac.uk/~mjw/pub/imas/
The possible interactions between meta-level control and deliberation are:

<table>
<thead>
<tr>
<th>Situation</th>
<th>Choose to Change Number of deliberations?</th>
<th>Changed intentions?</th>
<th>Would have reconsidered?(optimal?)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>-</td>
<td>No</td>
</tr>
</tbody>
</table>

**Situation Chosen to Change Number of deliberations?**

1. Yes, deliberations.
2. No, deliberations.
3. Yes, no deliberations.
4. No, no deliberations.

The possible interactions between meta-level control and deliberation are:
In situation (1), the agent did not choose to deliberate, and as a consequence, did not choose to change intentions. Moreover, if it had chosen to deliberate, it would not have changed intentions. An important assumption: cost of reconsider is much less behaving optimally.

In situation (2), the agent did not choose to deliberate, but did not change intentions. In this situation, the reconsider function is not behaving optimally.

In situation (3), the agent chose to deliberate, but did not change intentions. In this situation, the reconsider function is not behaving optimally.

In situation (4), the agent chose to deliberate, and did change intentions. In this situation, the reconsider function is behaving optimally.

An important assumption: cost of reconsider is much less than the cost of the deliberate process itself.
Kinny and Georgeff’s experimentally investigated effectiveness of intention reconsideration strategies. Two different types of reconsideration strategy were used:

- **bold agents** never pause to reconsider intentions, and
- **cautious agents** stop to reconsider after every action.

**Dynamism in the environment is represented by the rate of world change, γ.**
Results:

- If $\gamma$ is low (i.e., the environment changes slowly), then cautious agents tend to outperform bold agents. This is because they are able to recognize when situations and new opportunities arise.

- If $\gamma$ is high (i.e., the environment changes frequently), then bold agents are busy working towards achieving their intentions — and achieving — their intentions. Because cautious agents waste time reconsidering their commitments while bold agents do well compared to cautious ones. This is because they cannot change quickly.

- If $\gamma$ is low (i.e., the environment changes slowly), then cautious agents tend to outperform bold agents. This is because they are able to recognize when new opportunities arise.

An Introduction to Multiagent Systems

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
We now consider the semantics of BDI architectures to what extent does a BDI agent satisfy a theory of agency.

An Introduction to Multiagent Systems

Lecture 4

9 BDI Theory & Practice
From classical logic: \( \land, \lor, \neg \)

\[ \neg \]

### BDI Logic

- **Intends:** \( \phi \land (\phi \land \Box \phi) \)
- **Desires:** \( \phi \land \Box \phi \)
- **Believes:** \( \phi \land \Box \phi \)

### The BDI Connectives:

- \( \forall \) on all paths,
- \( \exists \) on some paths,

### The CTL *Path Quantifiers:

- \( \Box \) on all paths,
- \( \Diamond \) on some paths,
SemanticsofB-D-Icomponentsaregivenviaaccessibility
relationsover‘worlds’,whereeachworldisitselfa
branching
timestructure.
Propertiesrequiredofaccessibilityrelationsensurebeliefs
logic
KD45,desirelogicKD,intentionlogicKD.
(Plusinterreltionships...)

SemanticsofB-D-Icomponentsaregivenviaaccessibility
relationsover‘worlds’,whereeachworldisitselfa
branching
timestructure.
Let us now look at some possible axioms of BDI logic, and see to what extent the BDI architecture could be said to satisfy these axioms.

- Let \( \varphi \) be an arbitrary formula.
- \( \alpha \) be an \( O \)-formula, i.e., one which contains no positive occurrences of \( A \).

In what follows, let
Belief-goal compatibility:

$\text{Bel}(\alpha) \Leftrightarrow \text{Des}(\alpha)$

Goal-intention compatibility:

$\text{Int}(\alpha) \Leftrightarrow \text{Des}(\alpha)$

Operationalized in the $\text{deliberate}$ function.

Goal-intention compatibility:

States that having an intention to optionally achieve something implies having it as a goal (i.e., there are no intentions that are not goals).

This axiom is operationalized in the function $\text{options}$: an option should not be produced if it is not believed possible.

Operationalized in the $\text{deliberate}$ function.
Lecture 4: An Introduction to Multiagent Systems

Volitional commitment: If you intend to perform some action next, then you do a next.

Operationalized in the execute function.

Awareness of goals & intentions: Requires that new intentions and goals be posted as events.

Awareness of goals & intentions: ●

Volitional commitment: ●

\[
\begin{align*}
\phi & \iff \text{Bel} \phi \\
\phi & \iff \text{Des} \phi
\end{align*}
\]

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
An agent will eventually either act for an intention, or else drop it.

\[(\phi \land \lnot \text{Int}) \leftarrow (\phi)\]

- **No infinite deferral:**
  - Action to be posted.

A stronger requirement would be for the success or failure of the action.

Operationaized in the execute function.

If an agent does some action, then it is aware that it has done.

\[((\forall) \text{ done}) \leftarrow (\forall)\]

- **No unconscious actions:**
IRMA has four key symbolic data structures:

1. Implemented BDI Agents: IRMA

- beliefs: explicit representations of what is known and available to the agent
- plan library, and
- goals: things that the agent would like to make true — things that the agent has been allocated. In humans, not necessarily logically consistent, but our agents will be (goals)!

- desires: desires that the agent has chosen and committed to.

- intentions: desires that the agent has chosen and

IRMA has four key symbolic data structures:
Additionally, the architecture has:

- a deliberation process responsible for deciding upon the 'best' intentions to adopt.
- a filtering process determines which options are compatible with current intentions; and
- a means-ends analyzer determines which plans might be used to achieve intentions;
- an opportunity analyzer monitors the environment, and as a result of changes, generates new options;
- a reasoner for reasoning about the world; an inference engine;

An Introduction to Multiagent Systems
An Introduction to Multiagent Systems

- Intentions
- Beliefs
- Alternatives
- Filter
- Desires
- Filter
- Intentions
- Refinement
- Options
- Action
We now make the discussion even more concrete by introducing an actual agent architecture: the PRS.

In the PRS, each agent is equipped with a plan library.

The options available to an agent are directly determined by the plan library an agent has: an agent with no plans has no options.

In addition, PRS agents have explicit representations of beliefs, desires, and intentions, as above.

In addition, PRS agents have explicit representations of procedures.

About the mechanisms that can be used by the agent in order to realise its intentions.

In the PRS, each agent is equipped with a plan library.

12 Implemented BDI Agents: PRS
Vere & Bickmore developed HOMER: a simulated submarine, in a two-dimensional ‘Seaworld’.

HOMER — An Agent That Plans

HOMER takes instructions from a user in a subset of English with about 800 word vocabulary. HOMER can plan how to achieve its instructions, (which usually relate to collecting and moving items around the Seaworld) and can then execute its plans, modifying them as required during execution.

HOMER has a limited episodic memory, and using this, it is able to answer questions about its past experiences.
Lecture 4: An Introduction to Multiagent Systems

- simulator
- world
- text
- memory
- planner
- dynamic plan
- interpreter
- generator process
- reflective process
- episodic memory
- memory
- human director
- world simulator

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
EXAMPLE DIALOGUE 1

**U:** What is in front of you?

**H:** A log.

**U:** Do you own the log?

**H:** No, I don’t.

**U:** The log belongs to you.

**H:** Oh.

[LOG IS MOVED]

**U:** Turn away from your log.

**H:** OK, I am turning.

[LOG IS MOVED]

**U:** Turn around.

**H:** OK, I am turning.

**U:** Turn away from your log.

[...]

**H:** Oh.

**U:** The log belongs to you.

**H:** No, I don’t.

**U:** Do you own the log?

**H:** A log.

**U:** What is in front of you?
EXAMPLE DIALOGUE 2

U: Drop the package at the barge next Saturday at 9pm.

H: OK.

U: Are you going to the barge next Saturday?

H: Yes.

U: What time?

H: 8.56pm

[U: next Saturday?

H: Are you going to the barge next Saturday?

U: Drop the package at the barge.

[WHICH HOMER DETECTS]

USER ADDS CONTRADICTORY INSTRUCTIONS.

H: OK.

next Saturday at 9pm.

U: Drop the package at the barge.

[U: next Saturday?

H: Are you going to the barge next Saturday?

U: Drop the package at the barge.

[WHICH HOMER DETECTS]

USER ADDS CONTRADICTORY INSTRUCTIONS.

H: Yes.

U: What time?

H: 8.56pm

[U: next Saturday?

H: Are you going to the barge next Saturday?

U: Drop the package at the barge.

[WHICH HOMER DETECTS]

USER ADDS CONTRADICTORY INSTRUCTIONS.

H: OK.

next Saturday at 9pm.

U: Drop the package at the barge.

[WHICH HOMER DETECTS]

USER ADDS CONTRADICTORY INSTRUCTIONS.

H: Yes.

U: What time?

H: 8.56pm