CHAPTER 4: PRACTICAL REASONING AGENTS

An Introduction to Multiagent Systems

http://www.csc.liv.ac.uk/~mjw/pubs/tmas/
Practical reasoning is reasoning directed towards actions— the process of figuring out what to do.

Practical reasoning is reasoning from theoretical beliefs.

Theoretical reasoning is directed towards beliefs.

Distinguish practical reasoning from theoretical reasoning.

Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes (Bratman).

What is Practical Reasoning?
The ComponentsofPracticalReasoning

- Human practical reasoning consists of two activities:
  - deliberation
  - means-ends reasoning

  - deliberation — the outputs of deliberation are intentions.
  - means-ends reasoning — the outputs of means-ends reasoning are plans.

  deciding how to achieve these states of affairs
  deciding what state of affairs we want to achieve
Intentions in Practical Reasoning

1. Intentions pose problems for agents, who need to determine ways of achieving them.
2. Intentions provide a "filter" for adopting other intentions, which must not conflict.

If I have an intention to \( \phi \), you would not expect me to adopt an intention \( \psi \) that was incompatible with \( \phi \).

If I have an intention to \( \phi \), you would expect me to devote resources to deciding how to bring about \( \phi \).
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 being equal, it will try an alternative plan to achieve $\phi$.

4. Agents believe their intentions are possible. That is, they believe there is at least some way that the intentions could be brought about.
Agents do not believe they will not bring about their intentions.

5. Under certain circumstances, agents believe they will succeed with $\phi$.

6. Under certain circumstances, agents believe that under "normal" circumstances they would succeed with $\phi$.

If I intend $\phi$, then I believe that under "normal" circumstances I will bring about their intentions.

I would not be rational of me to adopt an intention to $\phi$ if I believed I would fail with $\phi$. 

An Introduction to Multiagent Systems 2e

Chapter 4
Agents need not intend all the expected side effects.

If I believe $\phi \Rightarrow \psi$ and I intend that $\phi$, I do not necessarily intend $\psi$ also. (Intentions are not closed under implication.) This last problem is known as the side effect or packaged deal problem.

An Introduction to Multiagent Systems
My desire to play basketball this afternoon is merely a potential influencer of my conduct this afternoon. If I intend to play basketball this afternoon, the matter is settled: I normally need not continue to weigh the pros and cons. When the afternoon arrives, I will normally just proceed to execute my intentions. (Bratman, 1990)

Intentions are Stronger than Desires
Planning is the design of a course of action that will achieve some desired goal.

Basic idea is to give a planning system:

- (representation of) goal/intention to achieve;
- (representation of) actions it can perform; and
- (representation of) environment;

and have it generate a plan to achieve the goal.

This is automatic programming.

Means-ends Reasoning/Planning
Plan to achieve goal

possible actions

environment

state of task

intention/goal
• plan itself.
• actions available to agent;
• state of environment;
• goal to be achieved;

Question: How do we represent...

Representations
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.

\[
\begin{align*}
\text{Holding}(x) & \quad \text{arm is holding } x \\
\text{Clean}(x) & \quad \text{nothing is on top of } x \\
\text{OnTable}(x) & \quad \text{obj} \ y \ \text{on the table} \\
\text{On}(x, y) & \quad \text{obj} \ y \ \text{on top of } x
\end{align*}
\]
Here is a representation of the blocks world described above:

- Clear(A)
- On(OnTable(C), A)
- OnTable(B)
- On(A, B)
- Clear(A)

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

- { OnTable(A), OnTable(B), OnTable(C) }

Here is a goal:

- A goal is represented as a set of formulae.
Actions in the STRIPS Representation

Each action has:

- An **add list** – list of facts made true by executing the action.
- A **delete list** – list of facts that are no longer true after action is performed.
- A **pre-condition list** – list of facts which must be true for action to be executed.
- A **name** – which may have arguments.
Example 1: The stack action occurs when the robot arm places the object \( x \) it is holding on top of object \( y \).

The stack action is defined as:

\[
\begin{align*}
\text{Stack}(x, y) & \equiv \text{add}\ (\text{ArmEmpty} \land \text{On}(x, y)) \\
& \land \text{del}\ (\text{Clear}(y) \land \text{Holding}(x)) \\
& \land \text{pre}\ (\text{Clear}(y) \land \text{Holding}(x))
\end{align*}
\]
Example 2: The unstack action occurs when the robot arm picks an object \( x \) up from on top of another object \( y \).

The stack and unstack actions are inverses of one another.

\[
\text{pre Unstack}(x, y) \equiv \neg \text{On}(x, y) \land \text{Clear}(x) \land \text{ArmEmpty} \\
\text{del On}(x, y) \land \text{Clear}(x) \land \text{ArmEmpty} \\
\text{add Holding}(x) \lor \text{Clear}(x) \\
\neg \text{ArmEmpty} \\
\neg \text{Holding}(x) \lor \text{Clear}(y) \\
\neg \text{On}(x, y) \\
\neg \text{ArmEmpty}
\]

\text{Stack and Unstack are inverses of one another.}
Example 3:
The *pickup* action occurs when the arm picks up an object $x$ from the table.

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

The putdown action occurs when the arm places the object \( x \) onto the table.
What is a plan?

A sequence (list) of actions, with variables replaced by constants.

•
A first pass at an implementation of a practical reasoning agent:

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).

Implementing Practical Reasoning Agents
• Problem: deliberation and means-ends reasoning processes are not instantaneous.

• They have a time cost.

• But the world may change.
Let's make the algorithm more formal.

1. \( B := B_0 \); /* initial beliefs */
2. while true do
   3. get next percept \( \rho \);
   4. \( B := \text{brf}(B, \rho) \);
   5. \( I := \text{deliberate}(B) \);
   6. \( \text{plan} := I \)
   7. \( \text{execute}(\text{plan}, B) \)
   8. end while
Deliberation

• How does an agent deliberate?

Chosen options are then intentions.

– choose between them, and commit to some;
– begin by trying to understand what the options available to you are;
– begin by trying to understand what the options available to you are;

• How does an agent deliberate?
The deliberate function can be decomposed into two distinct functional components:

- Option generation
- Filtering

Deliberation
In which the agent generates a set of possible alternatives, which takes the agent’s current beliefs and current intentions, and from them determines a set of options.

Represent option generation via a function, \( \text{options} \), which determines a set of desires.

\( \text{options} = \text{desires} \).
An agent uses a filter function.

In order to select between competing options, an alternative, and commits to achieving them.

In which the agent chooses between competing options.
11. end while
10. execute(π)
  9. plan(B, I)!
  8. filter(B, D, I)!
  7. options(B, I)!
  6. filter(B, D)!
  5. get next percept p!
  4. while true do
    3. I := I₀
    2. B := B₀
    1.
Undercommitment:

Commitment Strategies
After retrofitting, \( \text{Willie} \) is returned, marked "The Committed Assistant." You say "Never mind."

\[ \text{One minute later, \( \text{Willie} \) trundles over with a Genessee in its gripper. You say "Never mind."} \]

Then you ask: "What kind of beer did you buy?" It answers: "Genessee." Again, you ask \( \text{Willie} \) to bring you a beer. Again, it accedes, replying, "Sure."

\[ \text{After retrofitting, \( \text{Willie} \) is returned, marked "Model C: The Committed Assistant."} \]
Commitment became unachievable. By smashing the bottle, the commitment became unachievable. By smashing the bottle, the commitment became unachievable. By smashing the bottle, the commitment became unachievable. By smashing the bottle, the commitment became unachievable. By smashing the bottle, the commitment became unachievable. By smashing the bottle, the commitment became unachievable. By smashing the bottle, the commitment became unach...
Degrees of Commitment

- **Blind commitment**: A blindly committed agent will continue to maintain an intention until it believes that the intention has been achieved, or else that it is no longer possible to achieve the intention. Blind commitment is also sometimes referred to as fanatical commitment.

- **Single-minded commitment**: A single-minded 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.
An agent has commitment both to means and ends, i.e., the state of affairs, and the mechanism via which the agent wishes to achieve this. Currently, our agent control loop is overcommitted, i.e., replan if ever a plan goes wrong.

Modification: replan if ever a plan goes wrong.

Both to means and ends.

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
\begin{verbatim}
2. $B := B_0$;
3. $I := I_0$;
4. while true do
   5. get next percept $\rho$;
   6. $B := \text{brf}(B, \rho)$;
   7. $D := \text{options}(B, I)$;
   8. $I := \text{filter}(B, D, I)$;
   9. $\pi := \text{plan}(B, I)$;
   10. while not empty($\pi$) do
       11. $\alpha := \text{hd}(\pi)$;
       12. execute($\alpha$);
       13. $\pi := \text{tail}(\pi)$;
       14. get next percept $\rho$;
       15. $B := \text{brf}(B, \rho)$;
       16. if not sound($\pi, I, B$) then
           17. $\pi := \text{plan}(B, I)$;
   end-if
   end-while
end-while
\end{verbatim}
Still overcommitted to intentions: Never stops to consider whether or not its intentions are appropriate.

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

Single-minded commitment.
2. \( B := B_0 \);
3. \( I := I_0 \);
4. while true do
5. get next percept \( \rho \);
6. \( B := \text{brf}(B, \rho) \);
7. \( D := \text{options}(B, I) \);
8. \( I := \text{filter}(B, D, I) \);
9. \( \pi := \text{plan}(B, I) \);
10. while not empty(\( \pi \)) or succeeded(\( \pi \), \( I \), \( B \)) or impossible(\( \pi \), \( I \), \( B \)) do
11. \( \alpha := \text{hd}(\pi) \);
12. execute(\( \alpha \));
13. \( \pi := \text{tail}(\pi) \);
14. get next percept \( \rho \);
15. \( B := \text{brf}(B, \rho) \);
16. if not sound(\( \pi \), \( I \), \( B \)) then
17. \( \pi := \text{plan}(B, I) \);
Intention Reconsideration

• Our agent gets to reconsider its intentions once every iteration of the outer control loop, i.e., when:
  - it has completely executed a plan to achieve its current intentions;
  - it believes its current intentions are no longer possible;
  - it believes it has achieved its current intentions;
  - it has completely executed a plan to achieve its current intentions; or
  - it has completely executed a plan to achieve its current intentions; or
  - it has completely executed a plan to achieve its current intentions; or

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

• Modification: Reconsider intentions after executing every action.
2. \( B \) := \( B_0 \);
3. \( I \) := \( I_0 \);
4. while true do
5. get next percept \( \rho \);
6. \( B \) := \text{brf}(B, \rho);
7. \( D \) := \text{options}(B, I);
8. \( I \) := \text{filter}(B, D, I);
9. \( \pi \) := \text{plan}(B, I);
10. while not (\text{empty}(\pi) or \text{succeeded}(I, B) or \text{impossible}(I, B)) do
11. \( \alpha \) := \text{hd}(\pi);
12. execute(\alpha);
13. \( \pi \) := \text{tail}(\pi);
14. get next percept \( \rho \);
15. \( B \) := \text{brf}(B, \rho);
16. \( D \) := \text{options}(B, I);
17. \( I \) := \text{filter}(B, D, I);
18. if not \text{sound}(\pi, I, B) then
19. \( \pi \) := \text{plan}(B, I);
20. end if
21. end while
22. end while
An agent that constantly reconsider its attentions may spend insufficient time actually attempting to achieve its intentions, and hence runs the risk of never achieving them. An agent that does not stop to reconsider its intentions sufficiently often will continue attempting to achieve them even after it is clear that they cannot be achieved, or that there is no longer any reason for achieving them. But intention reconsideration is costly!

The Dilemma of Intention Reconsideration
Controlling Intention Reconsideration

- Solution: Incorporate an explicit meta-level control component that decides whether or not to reconsider.
2. $B := B_0$;
3. $I := I_0$;
4. while true do

\begin{align*}
5. & \text{get next percept } \rho; \\
6. & B := \text{brf}(B, \rho); \\
7. & D := \text{options}(B, I); \\
8. & I := \text{filter}(B, D, I); \\
9. & \pi := \text{plan}(B, I); \\
10. & \text{while not (empty(\pi) or succeeded(\pi) or impossible(\pi)) do}
\end{align*}

\begin{align*}
11. & \text{while true do}
\end{align*}

\begin{align*}
12. & \alpha := \text{hd}(\pi); \\
13. & \text{execute}(\alpha); \\
14. & \pi := \text{tail}(\pi); \\
15. & \text{get next percept } \rho; \\
16. & B := \text{brf}(B, \rho);
\end{align*}

\begin{align*}
17. & \text{if reconsider}(I, B) \text{ then}
\end{align*}

\begin{align*}
18. & D := \text{options}(B, I); \\
19. & I := \text{filter}(B, D, I);
\end{align*}

\begin{align*}
20. & \text{end-if}
\end{align*}

\begin{align*}
21. & \text{if not sound}(\pi, I, B) \text{ then}
\end{align*}

\begin{align*}
22. & \pi := \text{plan}(B, I);
\end{align*}

\begin{align*}
23. & \text{end-if}
\end{align*}

\begin{align*}
24. & \text{end-while}
\end{align*}
The possible interactions between meta-level control and deliberation are:

<table>
<thead>
<tr>
<th>Situation Choose to Changed Intention</th>
<th>Would have Changed Intention?</th>
<th>Number Deliberate Intention?</th>
<th>Optimal? Reconsider?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

The possible interactions between meta-level control and deliberation are:
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, $\gamma$. The effectiveness of intention reconsiderations strategies was experimentally investigated by Kinny and Georgeff. Two different types of reconsideration strategies were used:

- **Bold agents** never pause to reconsider intentions, and

- **Cautious agents** stop to reconsider after every action.

The rate of world change, $\gamma$, is used to represent the dynamism in the environment.
Results:

- If $\gamma$ is low (i.e., the environment does not change quickly), then bold agents do well compared to cautious ones. This is because cautious ones waste time reconsidering their commitments.

- If $\gamma$ is high (i.e., the environment changes frequently), then cautious agents tend to outperform bold agents. This is because they are able to recognize when intentions are doomed, and take advantage of serendipity.
We now make the discussion even more concrete by introducing an actual agent architecture: the PRS.

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

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

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

Implemented BDI Agents: PRS

An Introduction to Multiagent Systems 2e

Chapter 4

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
Chapter 4

An Introduction to Multiagent Systems
Example PRS (JAM) System

An Introduction to Multiagent Systems 2e

Chapter 4
EXECUTE print "World model at start is:\n";
ACHIEVE on "Block1" "Block2";
EXECUTE print "Achiieving Block1 on Block2.\n";
EXECUTE print "Achiieving Blocks on Block2.\n";
ACHIEVE on "Block2" "Block3";
EXECUTE print "Achiieving Block2 on Block3.\n";
ACHIEVE on "Block3" "Table";
EXECUTE print "Achiieving Blocks on Table.\n";
EXECUTE print "World model at end is:\n";
EXECUTE printWorldModel;

BODY:

CONTEXT:
ACHIEVE blocks-stacked;
GOAL:
"Establish Block1 on Block2 on Block3."
DOCUMENTATION:
NAME: "top-level plan"
PLAN:
Plan:

NAME: "Stack block that are already clear"

GOAL: ACHIEVE ON $OBJ1$ OBJ2;

CONTEXT:

BODY:

EXECUTE print "Making sure $OBJ1$ is clear
"

ACHIEVE CLEAR $OBJ1$

EXECUTE print "Making sure $OBJ2$ is clear.
"

ACHIEVE CLEAR $OBJ2$

EXECUTE print "Moving $OBJ1$ on top of $OBJ2$.
"

PERFORM move $OBJ1$ $OBJ2$;

UTILITY: 10;

FAILURE:

EXECUTE print "Stack blocks failed!
"

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

NAME: "Clearablock"

GOAL: ACHIEVE CLEAR OBJ;

CONTEXT: FACT ON OBJ2 OBJ;

BODY:

EXECUTE print "Clearing OBJ2 from on top of OBJ"

EXECUTE print "Moving OBJ2 to table"

ACHIEVE ON OBJ2 "Table"

EFFECTS:

EXECUTE print "Retracting ON OBJ2"

RETRACT ON OBJ1 OBJ

FAILURE:

EXECUTE print "Clearing block OBJ failed!"

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

NAME: Move a block onto another object

GOAL: PERFORM move OBJ1 OBJ2

CONTEXT: FACT CLEAR OBJ1
           FACT CLEAR OBJ2

BODY:
   EXECUTE print "Performing low-level move action"
   FACT ON OBJ1 OBJ3
   RETRACT ON OBJ1 OBJ2
   ASSERT CLEAR OBJ3
   ASSERT ON OBJ1 OBJ2

EFFECTS:
   WHEN: NOT (OBJ2 "Table")
          RETRACT CLEAR OBJ2
          FACT CLEAR OBJ3
          RETRACT CLEAR OBJ2;
   }
   }
   }
   }
   }

EXECUTE print "Move failed!"

FAILURES:

http://www.csc.liv.ac.uk/~mjw/pubs/imas/51
We now consider the semantics of BDI architectures: BDI Theory & Practice.

In order to give a semantics to BDI architectures, Rao & Georgeff have developed BDI logics: non-classical logics with modal connectives for representing beliefs, desires, and intentions.

de what extent does a BDI agent satisfy a theory of agency?
From classical logic: \( \land, \lor, \neg, \cdots \)

BDI Logic

• The BDI connectives:
  \( \phi \) (\( \phi \) intends)
  \( \phi \) (\( \phi \) desires)
  \( \phi \) (\( \phi \) believes)

• The CTL\( ^* \) path quantifiers:
  \( \phi \) (\( \phi \) on some paths\',
  \( \phi \) (\( \phi \) on all paths\',
  \( \exists \phi \) (\( \exists \phi \) on some paths\',
  \( \forall \phi \) (\( \forall \phi \) on all paths\',

BDILogic
Let us now look at some possible axioms of BDI logic,

\[ \forall \phi \] be an arbitrary formula.

In what follows, let

\[ \forall \alpha \text{-taiformula, i.e., one which contains no } \neg \alpha \text{-taiformula.} \]

\[ \forall \neg \alpha \text{-negative occurrence of } \neg \alpha. \]

Let us now look at some possible axioms of BDI logic,
Belief goal compatibility:

\( \text{Bel} \alpha \leftrightarrow (\text{Des} \alpha) \)

States that if the agent has a goal to optionally achieve something, this thing must be an option. This axiom is operationalized in the function \( \text{options} \): an option should not be produced if it is not believed possible. 
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).

Operationalized in the deliberate function:

\[(\text{Init } \alpha) \iff (\text{Des } \alpha)\]

An Introduction to Multiagent Systems 2e
Chapter 4
Volitional commitment: If you intend to perform some action next, then you do so next.

\[ \textit{Int does } (a) \iff (a) \textit{ does} \]

Operationalized in the \texttt{execute} function.

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
• Awareness of goals & intentions

\[
\begin{align*}
\text{Des} (\phi) & \Rightarrow \text{Bel}(\text{Int} (\phi)) \\
\text{Int} (\phi) & \Leftrightarrow \text{Bel}(\text{Des} (\phi))
\end{align*}
\]

Requires that new intentions and goals be posted as events.
If an agent does some action, then it is aware that it has done the action.

\[
\text{Bel}(\text{done}(a)) \leftrightarrow \text{done}(a)
\]

No unconscious actions.
An agent will eventually either act for an intention, or else drop it.

\((\phi \downarrow \text{Int} \leftarrow (\phi \leftarrow \text{Int} \downarrow ))\) 

---

No infinite deferral.