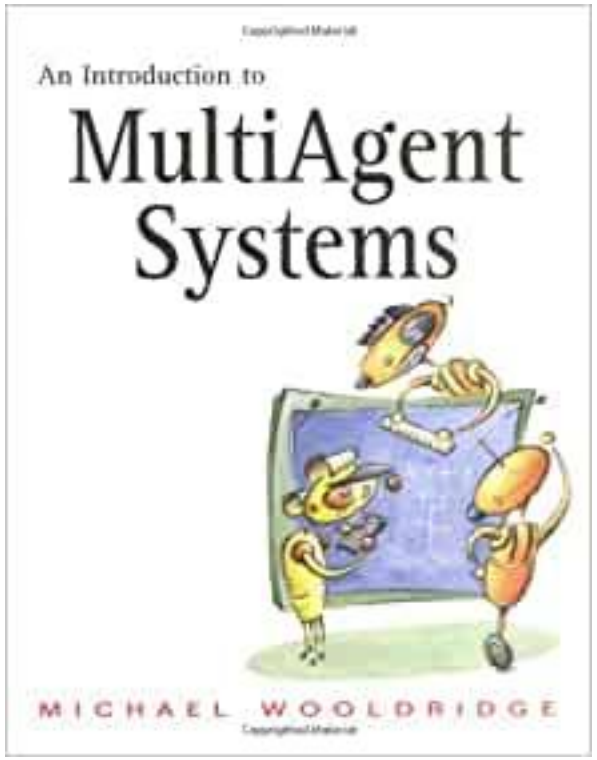
BDI Agents

Madalina Croitoru

University of Montpellier



Based on “An Introduction to MultiAgent Systems” by Michael Wooldridge, John Wiley & Sons, 2002.

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

Practical Reasoning

- Practical reasoning is reasoning directed towards actions — the process of figuring out what to do: “do I take the tram or the bike?”
  - Human practical reasoning consists of two activities:
    - *deliberation*
      deciding *what* state of affairs we want to achieve: “Tram or bike?” These are my **intentions**.
    - *means-ends reasoning*
      deciding *how* to achieve these states of affairs: “OK, I take the tram. What is the sequence of actions that I need to do?”
- Practical reasoning is distinguished from *theoretical reasoning* – theoretical reasoning is directed towards beliefs: “Tom is a cat, all cats are nice therefore Tom is nice.”

Intentions in Practical Reasoning

1. Agents need to determine ways of achieving intentions.
   If I have an intention to φ, you would expect me to devote resources to deciding how to bring about φ.
2. Intentions provide a "filter" for adopting other intentions, which must not conflict.
   If I have an intention to φ, you would not expect me to adopt an intention ψ such that φ and ψ are mutually exclusive.
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 φ fails, then all other things being equal, it will try an alternative plan to achieve φ.

Intentions and beliefs

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 need not intend all the expected side effects of their intentions.
   If I believe $\varphi \rightarrow \psi$ and I intend that φ, I do not necessarily intend ψ also. (Intentions are not closed under implication.)

   This last problem is known as the *side effect* or *package 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:

> "My desire to play basketball this afternoon is merely a potential influencer of my conduct this afternoon. It must vie with my other relevant desires […] before it is settled what I will do. In contrast, once I intend to play basketball this afternoon, the matter is settled […]. When the afternoon arrives, I will normally just proceed to execute my intentions." (Bratman, 1990)

What is Means-End Reasoning?

- Basic idea is to give an agent:
  - representation of goal/intention to achieve
  - representation of the actions it can perform
  - representation of the environment

  and have it generate a *plan* to achieve the goal
- Essentially, this is

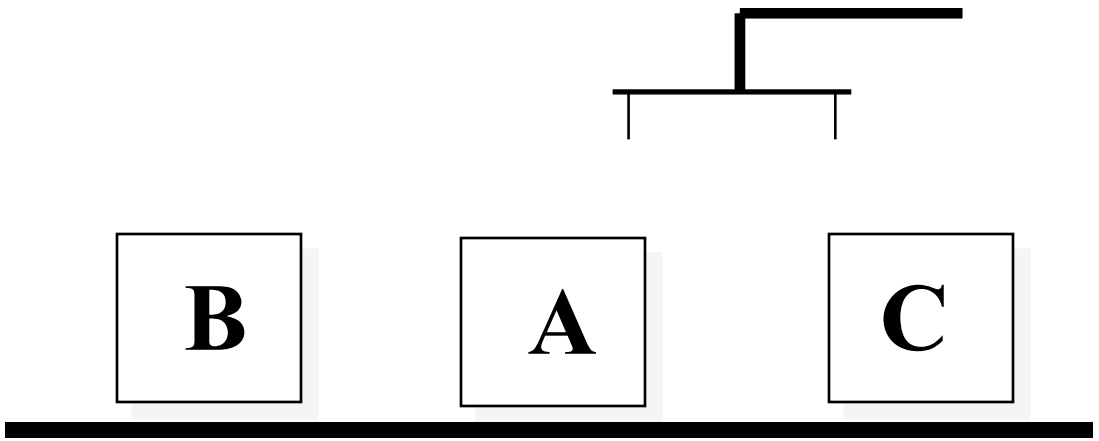
  automatic programming

Planning

- Question: How do we *represent*. . .
  - goal to be achieved
  - state of environment
  - actions available to agent
  - plan itself

The Blocks World

- A *goal* is represented as a set of formulae
- Here is a goal:
  $OnTable(A) \wedge OnTable(B) \wedge OnTable(C)$



The Blocks World

- Each action has:
  - a *name*
    which may have arguments
  - a *pre-condition list*
    list of facts which must be true for action to be executed
  - a *delete list*
    list of facts that are no longer true after action is performed
  - an *add list*
    list of facts made true by executing the action
- The *stack* action occurs when the robot arm places the object x it is holding is placed on top of object y.

$$
\begin{array}{ll}
 & Stack(x, y) \\
\text{pre} & Clear(y) \wedge Holding(x) \\
\text{del} & Clear(y) \wedge Holding(x) \\
\text{add} & ArmEmpty \wedge On(x, y)
\end{array}
$$

Planning

- Question: How do we *represent*. . .
  - ~~goal to be achieved~~
  - ~~state of environment~~
  - ~~actions available to agent~~
  - plan itself

The STRIPS approach

- The original STRIPS system used a goal stack to control its search
- The system has a database and a goal stack, and it focuses attention on solving the top goal (which may involve solving subgoals, which are then pushed onto the stack, etc.)

The Basic STRIPS Idea

- Place goal on goal stack:

- Considering top Goal1, place onto it its subgoals:

- Then try to solve subgoal GoalS1-2, and continue…

Implementing Practical Reasoning Agents

- 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))

Implementing Practical Reasoning Agents

- Let's make the algorithm more formal:

```
Agent Control Loop Version 2
1.   B := B_0; /* initial beliefs */
2.   while true do
3.        get next percept ρ;
4.        B := brf(B, ρ);
5.        I := deliberate(B);
6.        π := plan(B, I);
7.        execute(π)
8.   end while
```

Deliberation

- How does an agent deliberate?
  - begin by trying to understand what the *options* available to you are
  - *choose between them*, and *commit* to some
- Chosen options are then intentions

Deliberation

- The *deliberate* function can be decomposed into two distinct functional components:
  - *option generation*
    in which the agent generates a set of possible alternatives;
    Represent option generation via a function, *options*, which takes the agent's current beliefs and current intentions, and from them determines a set of options (= *desires*)
  - *filtering*
    in which the agent chooses between competing alternatives, and commits to achieving them.
    In order to select between competing options, an agent uses a *filter* function.

Deliberation

```
Agent Control Loop Version 3
1.
2.    B := B0;
3.    I := I0;
4.    while true do
5.          get next percept ρ;
6.          B := brf(B, ρ);
7.          D := options(B, I);
8.          I := filter(B, D, I);
9.          π := plan(B, I);
10.         execute(π)
11.   end while
```

Next time– more on deliberation (argumentation!)

- Exercise: Think about how to redo your implementation so far using the notions of belief, desire, intention and the algorithms presented in this course.
- For next session: upload a short video (less than 3 minutes), on YouTube preferably, of your planning agents so far.