BDI & INTRODUCTION TO JASON

BE4M36MAS - Multiagent systems
BELIEF-DESIRE-INTENTION
Model for programming autonomous agents using three concepts:

- Beliefs
- Desires
- Intentions
Beliefs

- agent's model of the world (what he supposes to be true)

Example:
- breeze(0, 1)
- stench(1, 0)
- pos(0, 0)
- safe(0, 0)
- safe(0, 1)
- safe(1, 0)
Beliefs

~ agent’s model of the world (what he supposes to be true)

Example:

breeze(0, 1). stench(1, 0).
pos(0, 0). safe(0, 0).
safe(0, 1). safe(1, 0).
Beliefs are not knowledge!

- An agent may believe facts that are not true.

Example:

Weather forecast announces nice weather for the weekend.

\[\text{nice_weather(sat). nice_weather(sun).}\]

→ You can believe that, but you cannot take it for granted.
Desires

Agent need not succeed in achieving all his desires, e.g.: Situation may not allow completing some of the desires. Desires may be mutually exclusive.
Desires

~ state of the world agent is **dreaming** about

Agent need not succeed in achieving all his desires, e.g.:
→ Situation may not allow completing some of the desires
→ Desires may be mutually exclusive
Intentions

Active goals of the agent (should not contradict beliefs)

Agent commits to fulfilling some of his desires. He must do everything he can to complete his intentions (unless specified otherwise).

Turning desires into intentions: deliberation

Realizing intentions: means-ends reasoning

Example: 

organize picnic(sat).
 Intentions

~ Active goals of the agent (should not contradict beliefs)

Agent commits to fulfilling some of his desires. He must do everything he can to complete his intentions (unless specified otherwise).

Turning desires into intentions: deliberation
Realizing intentions: means-ends reasoning

Example:

!organize_picnic(sat).
Commitments

Indicate that an agent has committed to some intention.

Optional: Situation in which an agent may forget about his intention (i.e. decommit)

Individual commitments

Social commitments
Commitments

~ indicate that an agent has committed to some intention

Optional: Situation in which an agent may forget about his intention (i.e. decommit)

• Individual commitments
• Social commitments
Individual commitments

- **Blind commitment** — the only way to decommit is to succeed
- **Single-minded commitment** — agent may decommit when he believes it is no longer possible to succeed
- **Open-minded commitment** — agent may decommit when he no longer believes it is possible to succeed
Example:

Recall \!\text{organize\_picnic}(\text{sat}). \text{and } \text{nice\_weather}(\text{sat}). Picnic can be organized only in good weather conditions.
Individual commitments

Example:

Recall !organize_picnic(sat). and nice_weather(sat).
Picnic can be organized only in good weather conditions.

- **Blind commitment**
**Example:**

Recall !organize_picnic(sat). and nice_weather(sat). Picnic can be organized only in good weather conditions.

- **Blind commitment** — Agent will be organizing the picnic event. Once he realizes that it’s raining whole Saturday, he crashes.
Example:

Recall $\text{organize\_picnic}(\text{sat}).$ and $\text{nice\_weather}(\text{sat}).$

Picnic can be organized only in good weather conditions.

- **Blind commitment** — Agent will be organizing the picnic event. Once he realizes that it’s raining whole Saturday, he crashes.

- **Single-minded commitment**
Individual commitments

Example:
Recall $\text{organize\_picnic(sat)}. \text{ and nice\_weather(sat)}$. Picnic can be organized only in good weather conditions.

- **Blind commitment** — Agent will be organizing the picnic event. Once he realizes that it’s raining whole Saturday, he crashes.

- **Single-minded commitment** — Agent will be organizing the event until rainy Saturday. He than resigns on his intention and the life goes by.
Individual commitments

Example:
Recall `!organize_picnic(sat). and nice_weather(sat)`. Picnic can be organized only in good weather conditions.

- **Blind commitment** — Agent will be organizing the picnic event. Once he realizes that it’s raining whole Saturday, he crashes.

- **Single-minded commitment** — Agent will be organizing the event until rainy Saturday. He than resigns on his intention and the life goes by.

- **Open-minded commitment**
Example:

Recall $\text{organize\_picnic}(\text{sat})$. and $\text{nice\_weather}(\text{sat})$. Picnic can be organized only in good weather conditions.

- **Blind commitment** — Agent will be organizing the picnic event. Once he realizes that it’s raining whole Saturday, he crashes.

- **Single-minded commitment** — Agent will be organizing the event until rainy Saturday. He then resigns on his intention and the life goes by.

- **Open-minded commitment** — Agent drops his intention as soon as the updated forecast is released.
JASON PROGRAMMING
Key components

- Set of beliefs (belief base)
- Set of intentions
Key components

- Set of beliefs (*belief base*)
- Set of intentions
- Plan library
- Set of events
Specifying initial beliefs and intentions

- **Specifying beliefs**
  - what agent knows at the beginning of the execution
    
    `depot(5,5).`
    `next(X,X+1).`

- **Specifying intentions**
  - what agent has to accomplish
    
    `!say_hello.`
    `!find_gold.`
    `!go_to(5,5).`
In order to execute an intention there must be an appropriate plan:

```r
say_hello.```

```r
print("Hello").```
In order to execute an intention there must be an appropriate plan:

```perl
+!say_hello <- .print("Hello").
```
Incorporating beliefs:

\[ \text{daytime(afternoon).} \]
\[ \text{!say_hello.} \]
Incorporating beliefs:

\[
\text{daytime(afternoon).} \\
\text{!say_hello.}
\]

\[
\text{+!say_hello : daytime(morning) \text{<- .print("Good morning").}} \\
\text{+!say_hello : daytime(afternoon) \text{<- .print("Good afternoon").}} \\
\text{+!say_hello : daytime(evening) \text{<- .print("Good evening").}}
\]
Incorporating variables (starting with **capital** letters):

```prolog
daytime(afternoon).
!say_hello("Bob").

+!say_hello(X) : daytime(T) <- .print("Good ", T, ", ", X).
```
Building plan database

+!say_hello(X) : daytime(T) <- .print("Good", T,"",",", X)

trigger  \hspace{1cm} context  \hspace{1cm} plan / subgoals

- **Trigger** — what event does the plan handle
- **Context** — condition when the plan is applicable
- **Plan** — what has to be done in order to fulfil the intention

Variables get **unified**, i.e. they get the value matching the intention specification / belief base.
Building plan database

- **Triggers**

<table>
<thead>
<tr>
<th></th>
<th>+ (additions)</th>
<th>– (removals)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intentions</td>
<td>+! intention(args)</td>
<td>–! intention(args)</td>
</tr>
<tr>
<td>Beliefs</td>
<td>+belief(args)</td>
<td>–belief(args)</td>
</tr>
</tbody>
</table>

- **Context** — Logical formula (using beliefs and percepts)
  
a & b, a | b, not a (belief a is not present), ~a

Percepts: pos(X,Y), cell(X,Y,T) (T=gold, depot, ally), carrying_gold, name(N), gsize(_,W,H)

- **Plan** — subgoals to achieve (separated by ;)
  
  - Intentions — !subgoal (sequential) / !!subgoal
  - Environment actions — do(left), do(right), do(up), do(down), do(pick), do(drop)
  - Internal actions — e.g. .print("Hello")
  - Belief base manipulation — +belief, –belief
EXAMPLE
Alice:

- During lunchtime, forward all calls to Carla.
- When I am busy, incoming calls from colleagues should be forwarded to Denise.
Example: Call forwarding rules

- During lunchtime, forward all calls to Carla.
  - What is lunchtime? — belief lunchtime(1130, 1230).
  - What is the incoming call? — addition of belief invite(X,Y)
Example: Call forwarding rules

• During lunchtime, forward all calls to Carla.
  • What is lunchtime? — belief lunchtime(1130, 1230).
  • What is the incoming call? — addition of belief invite(X,Y)
  • So...

  +invite(X,alice) : time(T) & lunchtime(S,E) & S<=T & T<=E <- !call_forward(alice, X, carla).
Example: Call forwarding rules

- When I am busy, incoming calls from colleagues should be forwarded to Denise.
  - What it means to be busy? — percept status(X, busy)
  - Who is a colleague? — belief colleague(X)
Example: Call forwarding rules

- When I am busy, incoming calls from colleagues should be forwarded to Denise.
  - What it means to be busy? — percept status(X, busy)
  - Who is a colleague? — belief colleague(X)
  - So...

    +invite(X, alice) :
      status(alice, busy) & colleague(X) <- !call_forward(alice, X, denise).
Example: Call forwarding rules

- Nothing prevents the connection:
  - How to make the connection? — Environment action
    \texttt{connect}(X,Y)
Example: Call forwarding rules

- Nothing prevents the connection:
  - How to make the connection? — Environment action
    connect(X,Y)
  - So...
    +invite(X,Y) : status(Y,idle) <- connect(X,Y).
Example: Call forwarding rules

- How to achieve the call forwarding?
Example: Call forwarding rules

- How to achieve the call forwarding?

  ```prolog
  +!call_forward(X,Y,Z) : status(Z, idle) <-
  -invite(X,Y) ; +invite(X,Z).
  ```