

BDI & INTRODUCTION TO JASON

AE4M36MAS - Multiagent systems

BELIEF-DESIRE-INTENTION

What is it?

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

Beliefs **are not** knowledge!

- An agent may **believe** facts that are **not true**.

Example:

Weather forecast announces nice weather for the weekend.

```
nice_weather(sat). nice_weather(sun).
```

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

~ 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).
```

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

Individual commitments

Example:

Recall `!organize_picnic(sat).` and `nice_weather(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** — Agent will be still organizing the picnic event. Once he realizes that it's raining whole Saturday, he crashes.

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 still 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.

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 still 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 simulation

```
depot(5,5).
```

```
next(X,X+1).
```

- **Specifying intentions**

~ what agent has to accomplish

```
!say_hello.
```

```
!find_gold.
```

```
!go_to(5,5).
```

`!say_hello.`

```
!say_hello.
```

In order to execute an intention there must be an appropriate **plan**:

```
+!say_hello <- .print("Hello").
```

Incorporating beliefs:

```
daytime(afternoon).  
!say_hello.
```

Incorporating beliefs:

```
daytime(afternoon).  
!say_hello.
```

```
+!say_hello : daytime(morning) <- .print("Good morning").  
+!say_hello : daytime(afternoon) <- .print("Good afternoon").  
+!say_hello : daytime(evening) <- .print("Good evening").
```

Incorporating variables (starting with **capital** letters):

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

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

$\underbrace{+!say_hello(X)}_{\text{trigger}} : \underbrace{daytime(T)}_{\text{context}} \leftarrow \underbrace{.print("Good ", T, ", ", ", X)}_{\text{plan / subgoals}} \quad \blacksquare$

- **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**

	+ (additions)	- (removals)
Intentions	<code>+! intention(args)</code>	<code>-! intention(args)</code>
Beliefs	<code>+belief(args)</code>	<code>-belief(args)</code>

- **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 `;`)

- Nesting intentions — `!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`