

Implementing the BDI Control Loop

Autonomous Software Agents

A.A. 2022-2023

Prof. Paolo Giorgini

Dr. Marco Robol



UNIVERSITY OF TRENTO - Italy

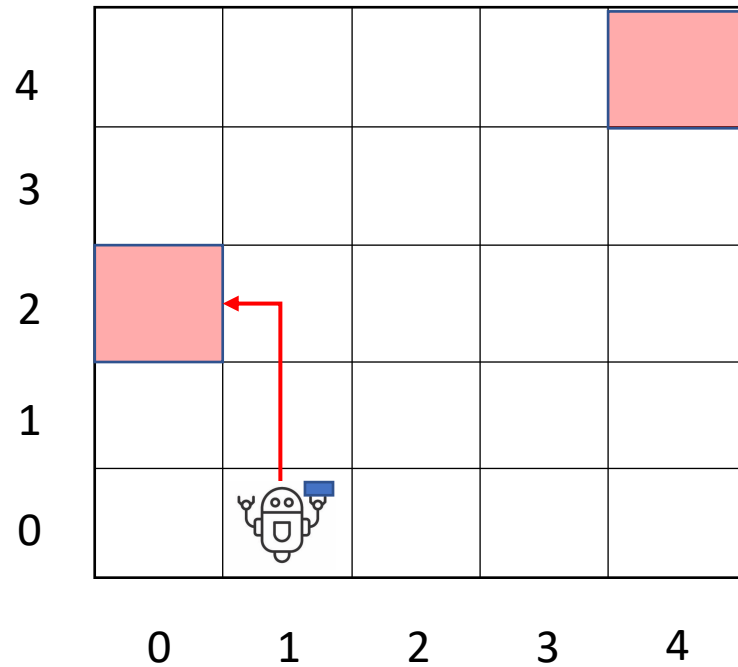
Department of Information
and Communication Technology

BDI control loop

- Given a set of beliefs **B**
 1. Decide about possible intentions to adopt
 2. Select new intentions to adopt
 3. Revise the Intention set **I**
 4. Revise and/or select new plans **P** for **I**
 5. Execute plans

Agent Control Loop Version 7

```
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.     while not ( $empty(\pi)$ 
11.                or  $succeeded(I, B)$ 
12.                or  $impossible(I, B)$ ) do
13.          $\alpha := hd(\pi);$ 
14.          $execute(\alpha);$ 
15.          $\pi := tail(\pi);$ 
16.         get next percept  $\rho;$ 
17.          $B := brf(B, \rho);$ 
18.         if  $reconsider(I, B)$  then
19.              $D := options(B, I);$ 
20.              $I := filter(B, D, I);$ 
21.         end-if
22.         if not  $sound(\pi, I, B)$  then
23.              $\pi := plan(B, I)$ 
24.         end-if
25.     end-while
26. end-while
```



```
-- t=0
B = {in(1,0), carry(pack_1), del_zone(0,2), del_zone(4,4)}
I = {}
P = {}
Do: Null
```

Intention Rules: if (prec) then Int

```
if (carry(Pck) ^ del_zone(X,Y)) { 0 += in(Pck,X,Y) }
```

```
→ 0={in(pack_1,0,2), in(pack_1,4,4)} # options
```

Selection: S = select(B,I,0)

```
→ S={in(pack_1,0,2)}. # seleted intentions
```

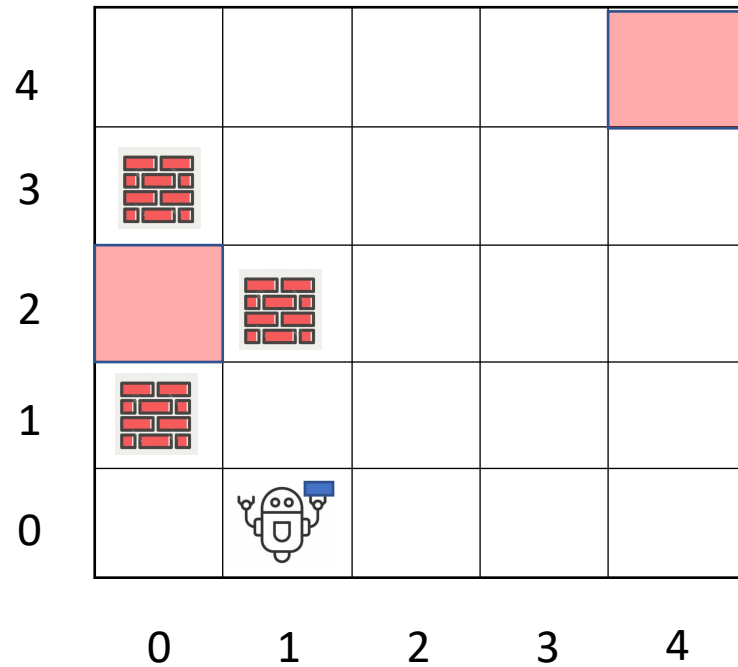
Intentions Revision: I = I_revision(B,I,S)

```
→ I={in(pack_1,0,2)} # new intentions
```

Plans selection or Plannig: P = planning (B,I,P)

```
→ P = {p1} # set of plans for I
```

```
→ p1 = {move(UP), move(UP), move(LEFT), put_down(pack_1)}
```



```
-- t=1
B = {in(1,0), carry(pack_1), del_zone(0,2), del_zone(4,4)}
I = {in(pack_1,0,2)}
P = {{move(UP), move(LEFT), put_down(pack_1)}}
Do: move(UP)
```

Options generation:

```
if (carry(Pck) ^ del_zone(X,Y)) { 0 += in(Pck,X,Y) }
```

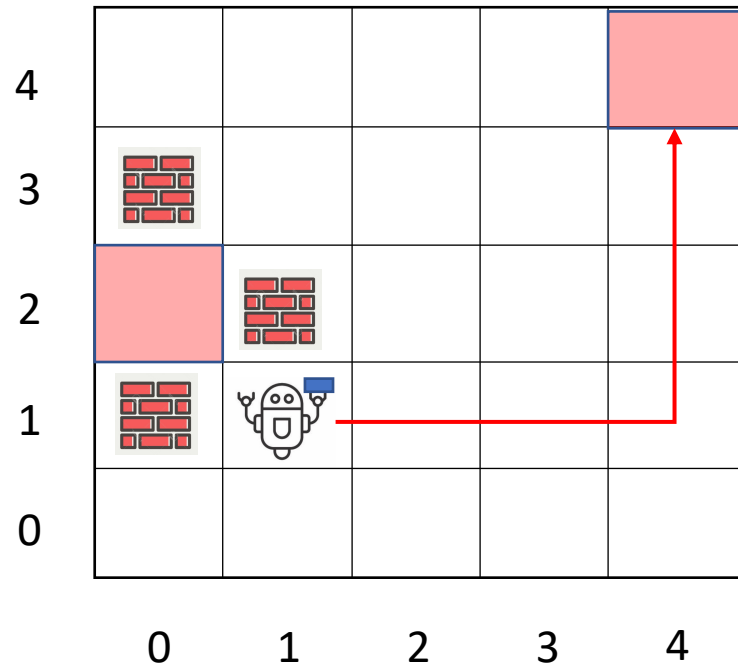
```
# again -> in(pack_1,0,2) and in(pack_1,4,4)
# but nothing changed from t=0 so it will be skipped
```

```
-- t=2
B = {in(1,1), carry(pack_1), del_zone(0,2), del_zone(4,4)
      in(block_1,0,1), in(block_2,1,2), in(block_3,0,3) }
I = {in(pack_1,0,2)}
P = {{move(LEFT), put_down(pack_1)}}
Do: move(UP)
```

Action failure: move(UP)

Re-Plannig: P = planning (B,I,P)

```
→ P = {} # no plans for in(pack_1,0,2)
```



Intentions Update: $I = I_update(B, I, \text{in}(\text{pack_1}, 0, 2))$

$\text{in}(\text{pack_1}, 0, 2)$ is not possible anymore

$\rightarrow I = \{\}$ # $\text{in}(\text{pack_1}, 0, 2)$ is dropped

Options generation:

if ($\text{carry}(\text{Pck}) \wedge \text{del_zone}(X, Y)$) { $0 += \text{in}(\text{Pck}, X, Y)$ }

again $\rightarrow \text{in}(\text{pack_1}, 0, 2)$ and $\text{in}(\text{pack_1}, 4, 4)$

in this case $\text{in}(\text{pack_1}, 0, 2)$ should not be considered

and only $\text{in}(\text{pack_1}, 4, 4)$ will be added

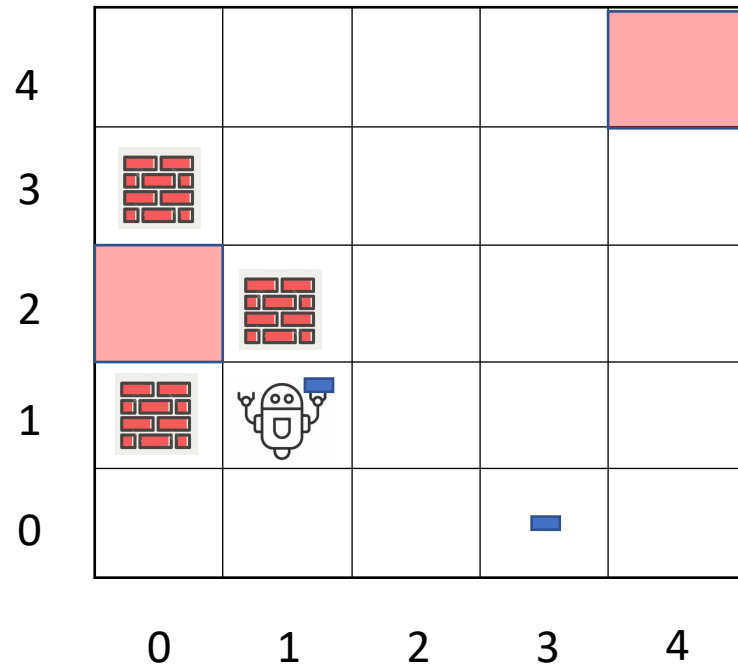
$\rightarrow 0 = \{\text{in}(\text{pack_1}, 4, 4)\}$ # options

Selection: $\rightarrow S = \{\text{in}(\text{pack_1}, 4, 4)\}$.

Intentions Revision: $\rightarrow I = \{\text{in}(\text{pack_1}, 4, 4)\}$

Plan selection:

$\rightarrow P = \{\{\text{move}(\text{RIGHT}), \text{move}(\text{RIGHT}), \text{move}(\text{RIGHT}), \text{move}(\text{UP}),$
 $\text{move}(\text{UP}), \text{move}(\text{UP}), \text{put_down}(\text{pack_1})\}\}$



with priority on Intentions

under the hypothesis that Intentions
cannot run in parallel

```
-- t=3
B = {in(1,1), carry(pack_1), del_zone(0,2), del_zone(4,4)
      in(block_1,0,1), in(block_2,1,2), in(block_3,0,3) }
I = {in(pack_1,4,4)}
P = {{move(RIGHT), move(RIGHT), move(UP),
      move(UP), move(UP), put_down(pack_1)}}}
Do: move(RIGHT)
```

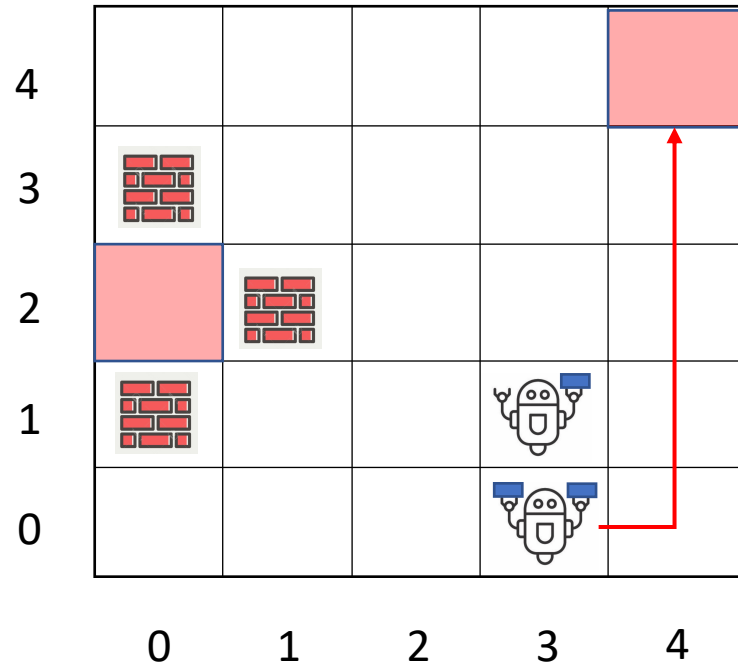
```
-- t=4
B = {in(2,1), carry(pack_1), del_zone(0,2), del_zone(4,4)
      in(block_1,0,1), in(block_2,1,2), in(block_3,0,3) }
I = {in(pack_1,4,4)}
P = {{move(RIGHT), move(UP), move(UP), move(UP),
      put_down(pack_1)}}}
Do: move(RIGHT)
```

```
-- t=5
B = {in(3,1), carry(pack_1), del_zone(0,2), del_zone(4,4)
      in(block_1,0,1), in(block_2,1,2), in(block_3,0,3),
      in(pack_2,3,0) }
I = {in(pack_1,4,4)}
P = {{move(UP), move(UP), move(UP), put_down(pack_1)}}}
Do: move(RIGHT)
```

Options generation + selection: $\rightarrow S=\{\text{pick_up}(\text{pack_2})\}$

Intentions Revision : $\rightarrow I=\{\text{carry}(\text{pack_2}), \text{in}(\text{pack_1},4,4)\}$

Re-planning: $\rightarrow I=\{\text{move}(\text{DOWN}), \text{pick_up}(\text{pack_2})\}$



-- t=6

B = {in(3,1), carry(pack_1), del_zone(0,2), del_zone(4,4)
in(block_1,0,1), in(block_2,1,2), in(block_3,0,3),
in(pack_2,3,0) }

I = {carry(pack_2), in(pack_1,4,4)}

P = {{pick_up(pack_2)}}}

Do: move(DOWN)

-- t=7

B = {in(3,0), carry(pack_1), del_zone(0,2), del_zone(4,4)
in(block_1,0,1), in(block_2,1,2), in(block_3,0,3),
in(pack_2,3,0) }

I = {carry(pack_2), in(pack_1,4,4)}

P = {}

Do: pick_up(pack_2)

-- t=8

B = {in(3,0), carry(pack_1), del_zone(0,2), del_zone(4,4)
in(block_1,0,1), in(block_2,1,2), in(block_3,0,3),
in(pack_2,3,0) }

I = {carry(pack_2), in(pack_1,4,4)}

P = {}

Do: NULL

-- t=9

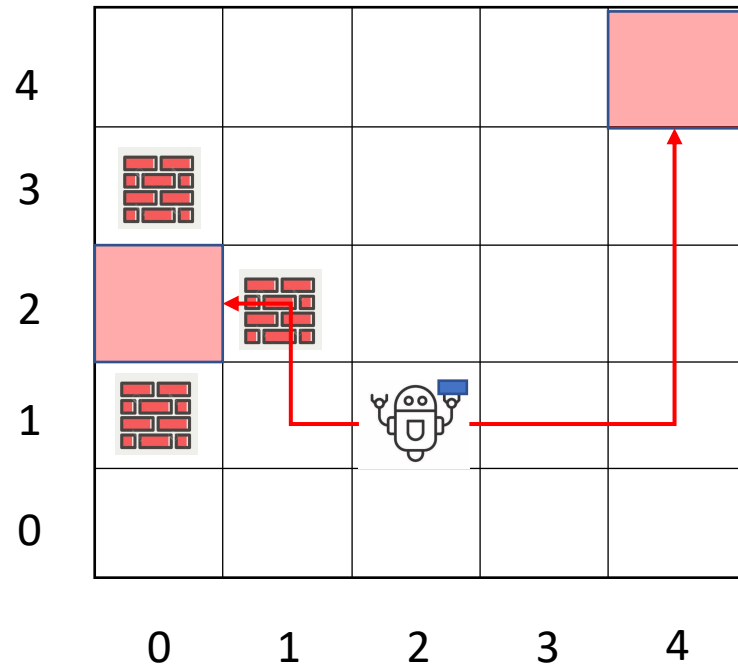
B = {in(3,0), carry(pack_1), del_zone(0,2), del_zone(4,4)
in(block_1,0,1), in(block_2,1,2), in(block_3,0,3),
carry(pack_2)}

I = {in(pack_1,4,4)}

P = {{move(UP), move(UP), move(UP), move(UP), put_down(pack_1)}}}

Do: move(RIGHT)

Re-Planning for the second intention



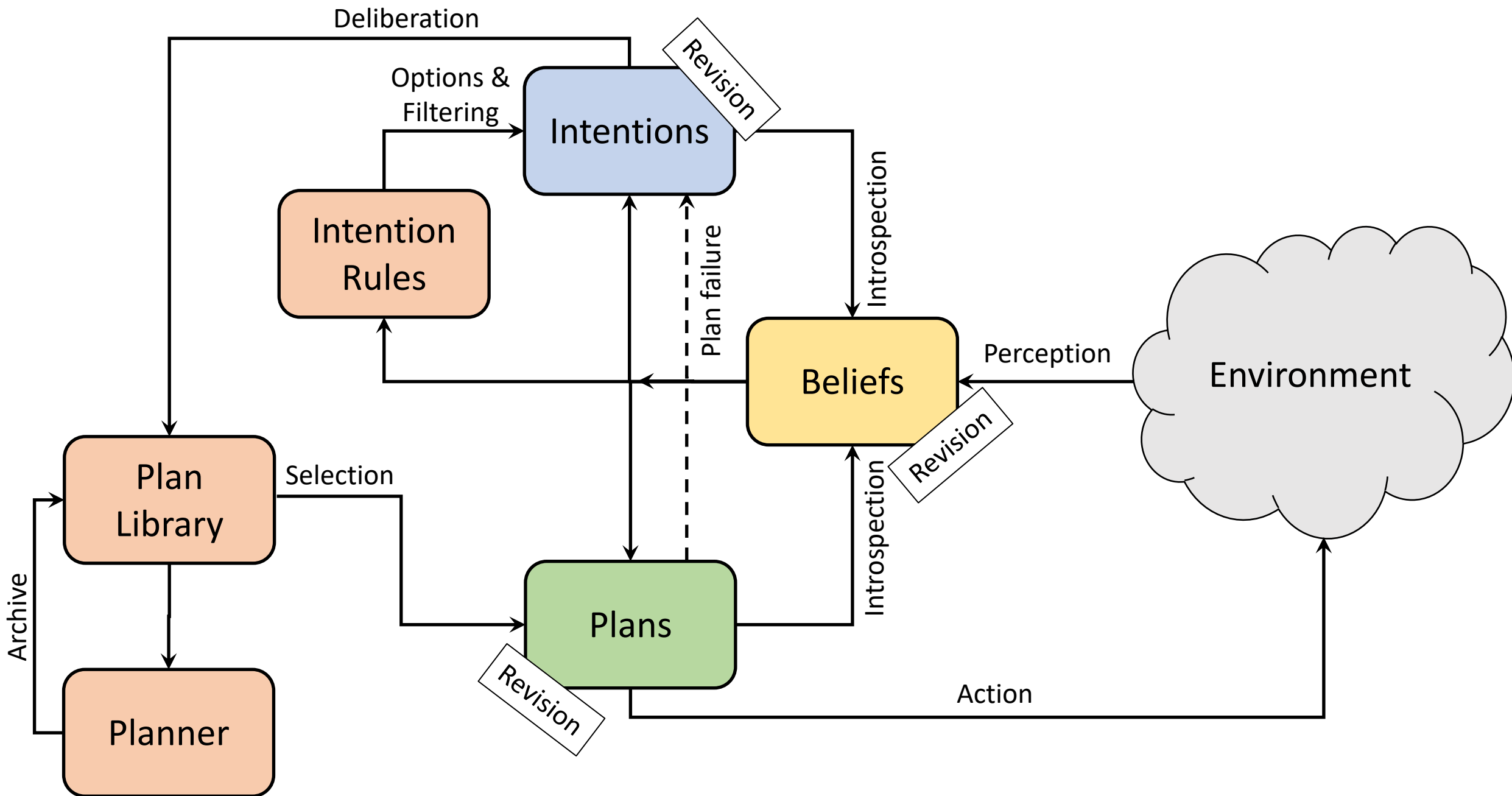
```
-- t=4
B = {in(2,1), carry(pack_1), del_zone(0,2), del_zone(4,4)
      in(block_1,0,1), in(block_3,0,3) }
I = {in(pack_1,4,4)}
P = {{move(RIGHT), move(UP), move(UP), move(UP),
      put_down(pack_1)}}
Do: move(RIGHT)
```

Options generation + selection: $\rightarrow S=\{\text{in}(\text{pack_1},0,2)\}$

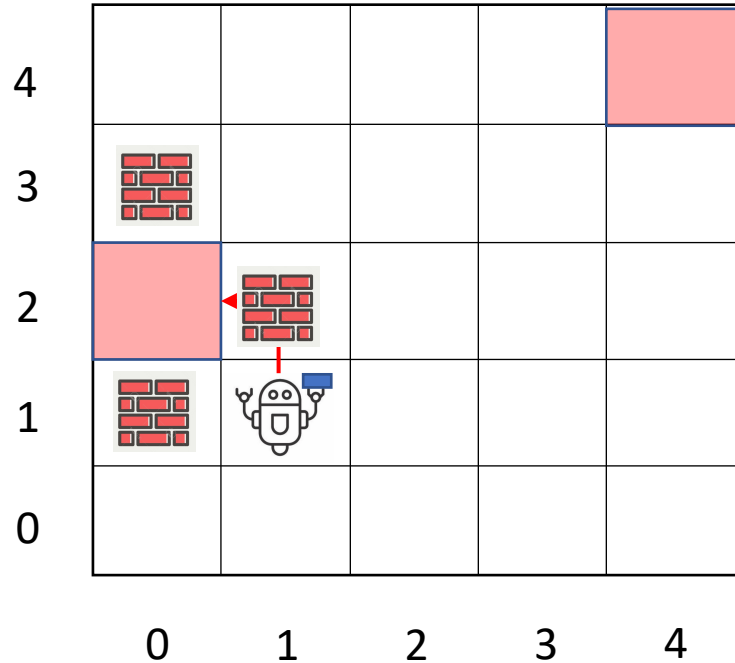
Intentions Revision : $\rightarrow I=\{\text{in}(\text{pack_1},0,2)\}$

Re-planning: $\rightarrow I=\{\text{move(LEFT), move(UP), move(LEFT),}$
 $\text{put_down(pack_q)}\}$

```
-- t=5
B = {in(2,1), carry(pack_1), del_zone(0,2), del_zone(4,4)
      in(block_1,0,1), in(block_3,0,3) }
I = {in(pack_1,0,2)}
P = {move(UP), move(LEFT), put_down(pack_q)}
Do: move(LEFT)
```

The order of things and timing



-- t=0

B = {in(1,1), carry(pack_1), del_zone(0,2), del_zone(4,4)}

I = {}

P = {}

Do: Null

Options + Filtering + Int. Revision + Planning

What about if in the meanwhile things in the environment change?

in(block_1,0,1), in(block_2,1,2), in(block_3,0,3)

-- t=1

B = {in(1,1), carry(pack_1), del_zone(0,2), del_zone(4,4),
in(block_1,0,1), in(block_2,1,2), in(block_3,0,3) }

I = {in(pack_1,0,2)}

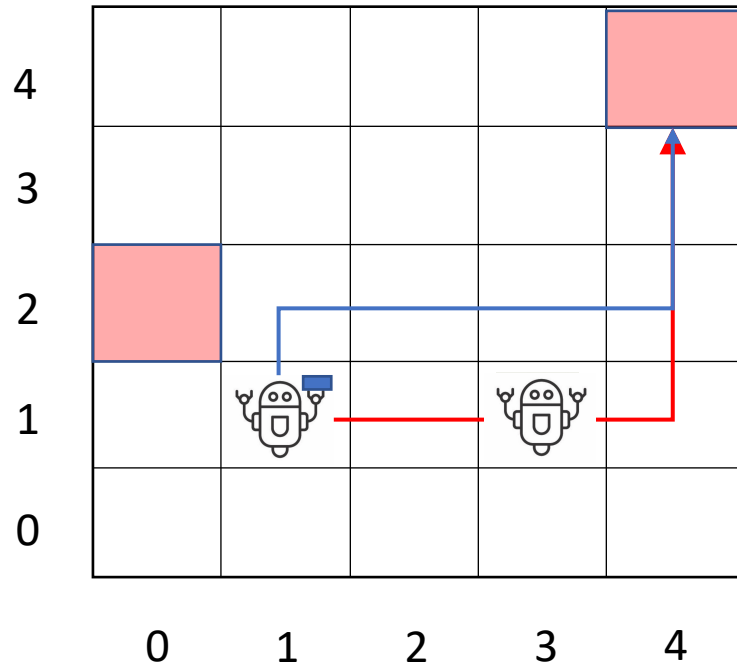
P = {{move(LEFT), put_down(pack_1)}}

Do: move(UP)

When do we check?

- Much it depends on the application and on the dynamicity of the environment
 - check beliefs at any decision point
 - Option / filtering / deliberation / planning / action
 - check soundness or optimality of the plan before any action
 - Always evaluate the best way to achieve a goal
 - revise intentions before any action
 - This may bring the agent to a schizophrenic behaviour but it could be useful to have agents that are opportunistic

Be careful



```
-- t=0
B = {in(1,1), carry(pack_1), del_zone(0,2), del_zone(4,4)}
I = {in(pack_1,4,4)}
P = {{move(RIGHT), move(RIGHT), move(UP), move(UP), move(UP),
put_down(pack_1)}}
Do: move(RIGHT)
```

The plan is not sound anymore: → **Re-planning**

since the block will remain in the position (3,1),
the plan is not sound now and in the future

The plan is not sound anymore: → **Re-planning ?**

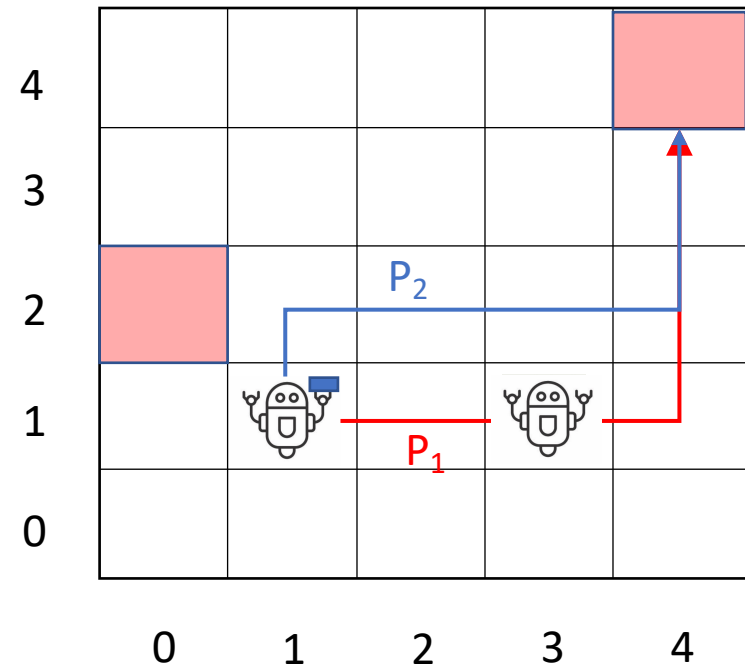
Will the agent remain in the position (3,1)?
if we replan, it may happen that the new plan will not
be sound anymore in the next step

The agent can distinguish between a block and a moving block

- Different strategies in case of moving blocks

Moving obstacles

- Consider moving obstacles as permanent obstacles
 - No differences in the behaviour
- Elaborating a probability model for the trajectory of the obstacle
 - The new plan will be selected on the base of the probability that the moving obstacle can make the plan not sound at some point (P_1 vs P_2)
- Other strategies
 - Eg. , stay away from other agents or techniques based on game theory



Using the probability model

Under the hypothesis of having the same probability that **Agent 1** move in any tile or stay where it is

The probability that **Agent 2** is an obstacle for **Agent 1** executing P_1 is:

- $P(\text{move}(\text{LEFT})) + P(\text{not move}) = 1/5 + 1/5 = 0.4$

The probability that **Agent 2** is an obstacle for **Agent 1** executing P_2 is:

-

