

Components of PDDL planning task

The planning task: Job acquisition facility for a factory

Objects: Things of interest

The workspace

- Order workspace: This serves as the workspace where the orders are placed.
- Delivery workspace: workspace where finished items are dropped for delivery to customers.
- Factory workspace: workspace where each factory operation carried by an agent is performed.
- Workstation: workstation where each operations is done inside the factory

The robots

- Transporter: transports items from one workspace/workstation to another
- fixed manipulators: pick and drop items from workspace to transporter and vice versa
- mobile manipulators: pick, drop and transports
- painter agent: color items
- Order agent: receives orders and check if the operation can be done
- Delivery agent: confirm delivery and check if all operations are correctly done.

The items

- Screw
- Nuts
- Bolts

IN PDDL:

```
(:objects workspace0 workspaceD workspaceF workstation  
      nut bolt  
      transporter manipulator mobile painter order  
      delivery)
```

Predicates: Depicts the objects properties or attributes which can be true or false
listed above are workspaces, workstation, robots, items

- Is x a workspace?
- Is x a robot?
- Is workstation empty?
- Is x an item?
- Is item x in workspace y?
- Is robot x in workspace y?
- Is robot x idle?
- Is color agent x painting?
- Is item x painted?
- Is transporter x transporting?
- Is manipulator x picking?
- Is manipulator x dropping?

- Is robot x free?
- Is robot x confirming orders?
- Is robot x confirming delivery?

WORKSPACE(x)---x is a workspace
 ROBOT(x)-----x is a robot
 ITEM(x)-----x is an item
 WORKSTATION(x)- x is a workstation
 at-item(x, y)-----x is an item and in workspace y
 at-robot(x, y)-----x is a robot and in workspace y
 idle(x)-----x is a robot and it is idle
 free(x)-----x is a robot and x does not hold an item
 carry(x, y)----- x is a robot , y is an item, and x holds y
 empty(x)-----x is a workstation and it is empty
 ontop(x, y)-----x is a robot, y is an item, item on transporter
 on(x, y)-----x is a robot, y is an item , item on workspace
 paint(x, y)-----x is a robot, y is an item and x paints y

In PDDL:

```
(:predicates (WORKSPACE ?x) (ITEM ?x) (ROBOT ?x)
              (at-item ?x ?y) (at-robot ?x ?y) (idle ?x)
              (free ?x) (carry ?x ?y) (ontop ?x ?y) (empty ?x)
              (on ?x ?y) paint(?x ?y) paint(?x))
```

Initial state : start state of the world

WORKSPACE(workspaceO), WORKSPACE(workspaceD) and WORKSPACE(workspaceF) are **true**

ITEM(screw), ITEM(nut) and ITEM(bolt) are **true**

ROBOT(manipulator), ROBOT(order), ROBOT(delivery), ROBOT(transporter), and ROBOT(mobile) , ROBOT(painter) are **true**.

idle(manipulator), idle(deilvery), idle(mobile), idle(transporter), idle(order), idle(painter) are **true**

at-robot(order, workspaceO), at-robot(manipulator, workspaceO), at-robot(mobile, workspaceO), at-robot(transporter, workspaceO), at-item(screw, workspaceO) are **true**

at-robot(painter, workspaceF), at-robot(manipulator, workspaceF), at-robot(transporter, workspaceO) are **true**

at-robot(delivery, workspaceD) are **true**
Every other things are **false**.

In PDDL:

```
(:init (WORKSPACE workspace0) (WORKSPACE workspaceF)
      (WORKSPACE workspaceD)
      (ITEM screw) (ITEM nut) (ITEM bolt)
      (ROBOT order) (ROBOT manipulator) (ROBOT transporter)
      (ROBOT mobile) (ROBOT delivery) (ROBOT painter)
      (idle order) (idle manipulator) (idle transporter)
      (idle mobile) (idle delivery) (idle painter)
      (at-robot order workspace0)
      (at-robot manipulator workspace0)
      (at-robot transporter workspace0)
      (at-robot mobile workspace0)
      (at-item nut workspace0)
      (at-robot color workspaceF)
      (at-robot manipulator workspaceF)
      (at-robot transporter workspaceF)
      (at-robot painter workspaceF)
      (at-robot delivery workspaceD))
```

Goal specifications:

at-item(screw, workspaceD) are **true**
Others remains unchange

In PDDL:

```
(:goal (and (at-item screw workspaceD) (paint screw)))
```

ACTIONS

Actions/Operator: move

Description: Any robot x can move from workspace y to workspace z

Precondition: WORKSPACE(y) , WORKSPACE(z), ROBOT(x) and at-robot(x, y) are **true**

Effects: at-robot(x, z) becomes true, at-robot(x, y) becomes **false**. **Everything else remains the same.**

Actions/Operator: pickup

Description: A robot x can pick up item y in workspace z

Precondition: ROBOT(x), ITEM(y), WORKSPACE(z), at-item(y, z), at-robot(x, z) and free(x) are **true**

Effects: carry(x, y) becomes **true**. At-item(y, z) and free(x) are **false**. Everything else remains the same.

Actions/Operator: paint

Description: robot x can paint item y

Precondition: ROBOT(x), ITEM(y), WORKSPACE(z), at-item(y, z), at-robot(x, z), and free(x) are **true**

Effects: paint(x, y) are **true**. free(x) are **false**. Everything else remains the same.

Actions/Operator: load

Description: robot x can load item y in workspace z on transporter

Precondition: ROBOT(x), ROBOT(x1), ITEM(y), WORKSPACE(z), at-robot(x, z), idle(x1) and free(x) are **true**

Effects: carry(x, y), empty(z) and ontop(y, x1) are **true**, idle(x1), free(x) are **false**. Everything else remains the same.

Actions/Operator: unload

Description: robot x can unload item y in workspace z on workstation w

Precondition: ROBOT(x), ROBOT(x1), ITEM(y), WORKSPACE(z), WORKSTATION(w), at-robot(x, z), at-robot(x1, z), empty(w), at-item(y, x1) and free(x) are **true**

Effects: carry(x, y) and on(y, w) are **true**. free(x), idle(x1) becomes **false**. Everything else remains the same.