Introduction to Cognitive Robotics

Module 11: Cognition-enabled Robot Manipulation with CRAM

Lecture 4: Defining a new grasp

David Vernon
Carnegie Mellon University Africa

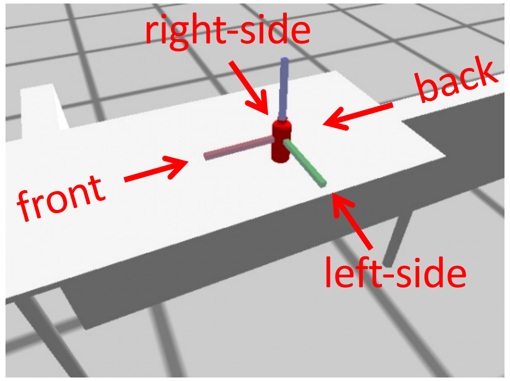
www.vernon.eu

Defining a New Grasp Pose

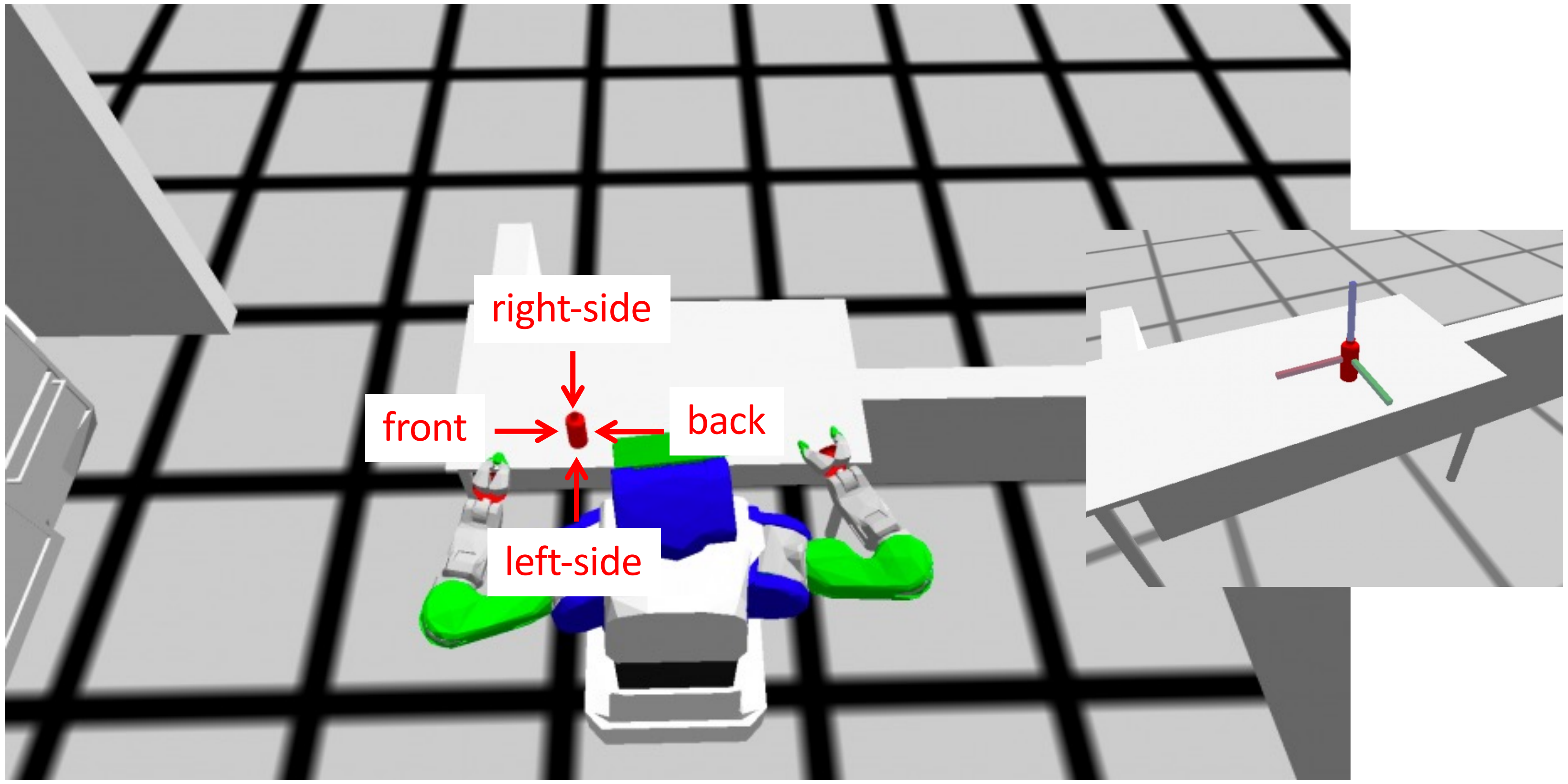
- A bottle has four pre-defined grasp poses associated with it

- Each defined with respect to the frame embedded in the bottle

front	grasp frame is aligned with the positive *X* axis, directed towards the frame origin
back	grasp frame is aligned with the negative *X* axis, directed towards the frame origin
left-side	grasp frame is aligned with the positive *Y* axis, directed towards the frame origin
right-side	grasp frame is aligned with the negative *Y* axis, directed towards the frame origin

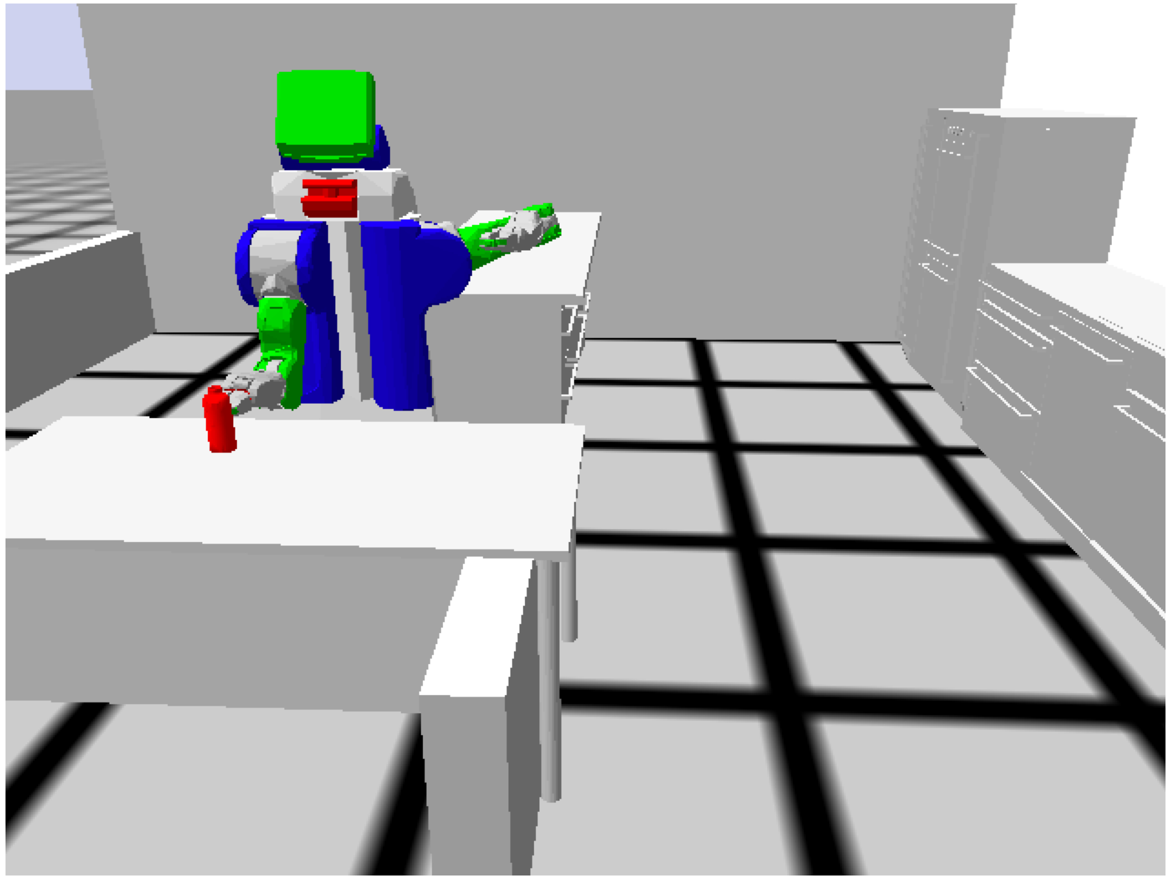


Defining a New Grasp Pose



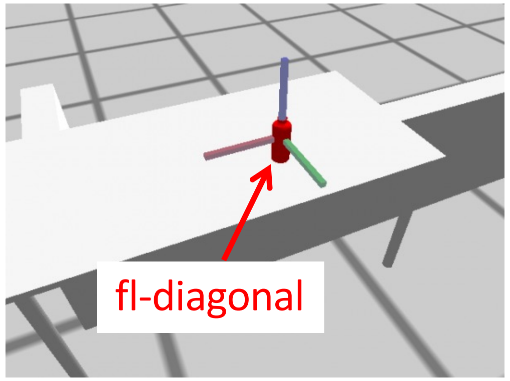
Defining a New Grasp Pose

This is grasping from the left-side



Defining a New Grasp Pose

- Now we explain how to define a new grasp pose and an approach poses
- Let's grasp the bottle diagonally between positive X and Y axes
- We'll call this the front-left-diagonal grasp: fl-diagonal



Defining a New Grasp Pose

The grasps are defined as `obj_T_grp`

- i.e. the coordinate frame of the gripper (or end-effector) with respect to the object (see CR04-01)
- All the grasp poses are defined in the object coordinate frame, as expected:
  - The origin of the object coordinate frame is defined to be the center of the bounding box of the object
  - The X axis is usually either
    - the longer principal axis of the object or
    - the axis from the handle towards the head of the object
  - The Z axis is usually defined to be directed upwards, with the object is oriented as it would be typically standing on a supporting surface
  - The Y makes us a right-hand system, as usual

Defining a New Grasp Pose

- Equivalently, the gripper pose `obj_T_grp`
  - is the pose of the tool frame (or tool centre point TCP) in the end-effector (see CR04-03)
- To define the grasp pose, we need to define
  - The translation the gripper with respect to the object frame
  - The orientation the gripper with respect to the object frame

Defining a New Grasp Pose

- To calculate the translation part, we'll define some offsets as parameters
- The values are taken from the predefined grasp offset values in workspace/ros/src/cram/cram_knowrob/cram_knowrob_pick_place/src/grasping.lisp

```
(defparameter *lift-z-offset* 0.15 "in meters")
(defparameter *lift-offset* `(0.0 0.0 ,*lift-z-offset*))

(defparameter *bottle-pregrasp-xy-offset* 0.15 "in meters")
(defparameter *bottle-grasp-xy-offset* 0.02 "in meters")
(defparameter *bottle-grasp-z-offset* 0.005 "in meters")
```

This is the translation vector that defines the origin of the lift pose after lifting the bottle

```
(defparameter *lift-z-offset* 0.15 "in meters")
(defparameter *lift-offset* `(0.0 0.0 ,*lift-z-offset*))

(defparameter *bottle-pregrasp-xy-offset* 0.15 "in meters")
(defparameter *bottle-grasp-xy-offset* 0.02 "in meters")
(defparameter *bottle-grasp-z-offset* 0.005 "in meters")
```

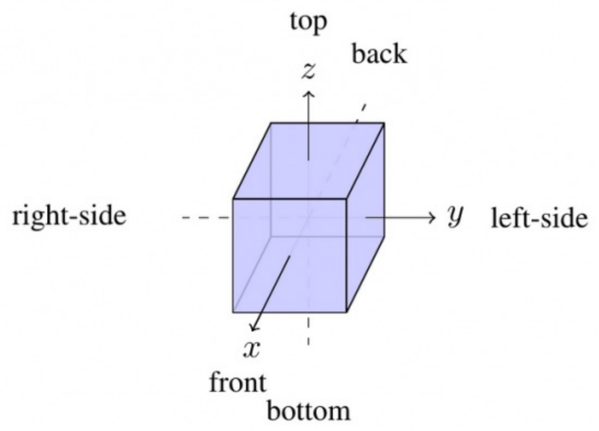
This value is used to define the origin of the pose before grasping: the pre-grasp pose (the x and y translation distances)

```lisp
(defparameter *lift-z-offset* 0.15 "in meters")
(defparameter *lift-offset* `(0.0 0.0 ,*lift-z-offset*))

(defparameter *bottle-pregrasp-xy-offset* 0.15 "in meters")
(defparameter *bottle-grasp-xy-offset* 0.02 "in meters")
(defparameter *bottle-grasp-z-offset* 0.005 "in meters")
```

These values are used to define the origin of the grasp pose

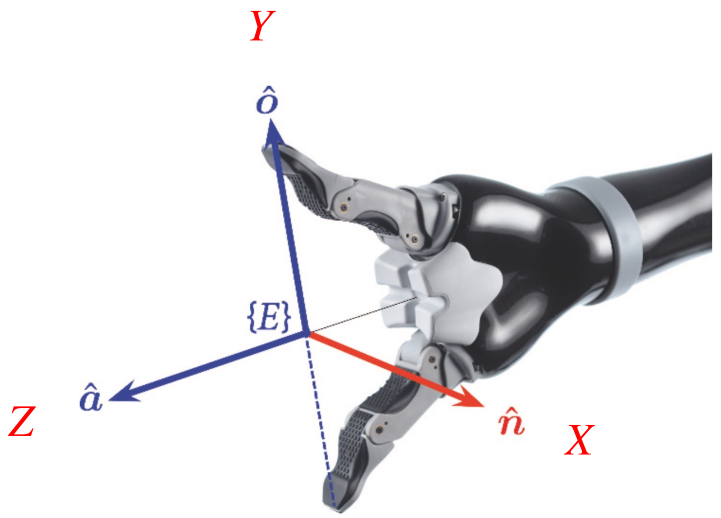
Defining a New Grasp Pose

- To calculate the orientation, we need to know
  - the coordinate frame of the object
  - the coordinate frame of gripper in order to define a grasp
- A front-left-diagonal grasp means that the gripper has to come from the positive X and Y axis side of the object



Recall from CR04-03

The same convention applies to the $\boldsymbol{E}$ frame that is embedded in a two-finger gripper (end-effector … hence $\boldsymbol{E}$)



(Corke, 2017), p. 41

$$E = \begin{array}{cccc} \text{Direction of } X \text{ axis} & \text{Direction of } Y \text{ axis} & \text{Direction of } Z \text{ axis} & \\ \downarrow & \downarrow & \downarrow & \\ \end{array}$$

$$E = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

n	Normal
o	Orientation
a	Approach

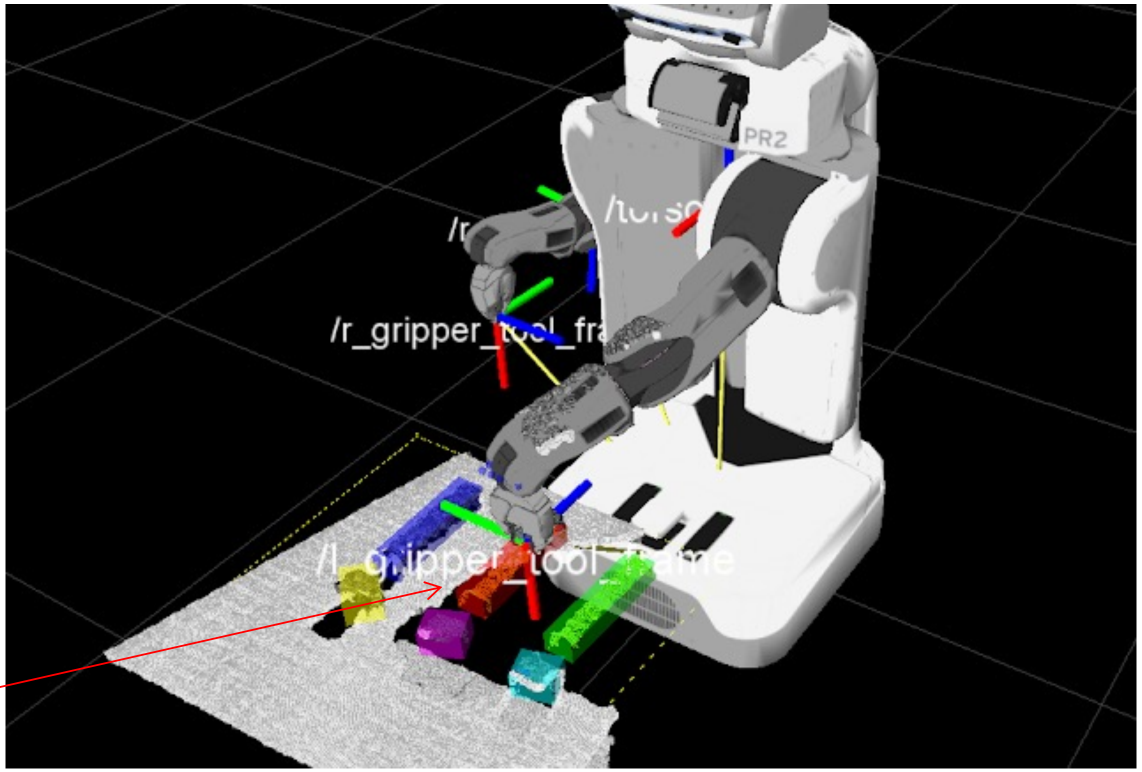
Recall from CR04-03

ROS uses a different convention

"If the end effector is a grasping device, the frame should be located at the recommended object grasping location. The frame orientation is defined as X the axis going 'toward' the object. Y the main dimension in which the grasping device moves and Z orthogonal to X and Y axes."

https://www.ros.org/reps/rep-0120.html#l-gripper-and-r-gripper

This approach is consistent with the convention of embedding a frame in a vehicle, with the X axis aligned with the direction of travel; see conventions on specifying orientation using roll, pitch, and yaw in the following slides.



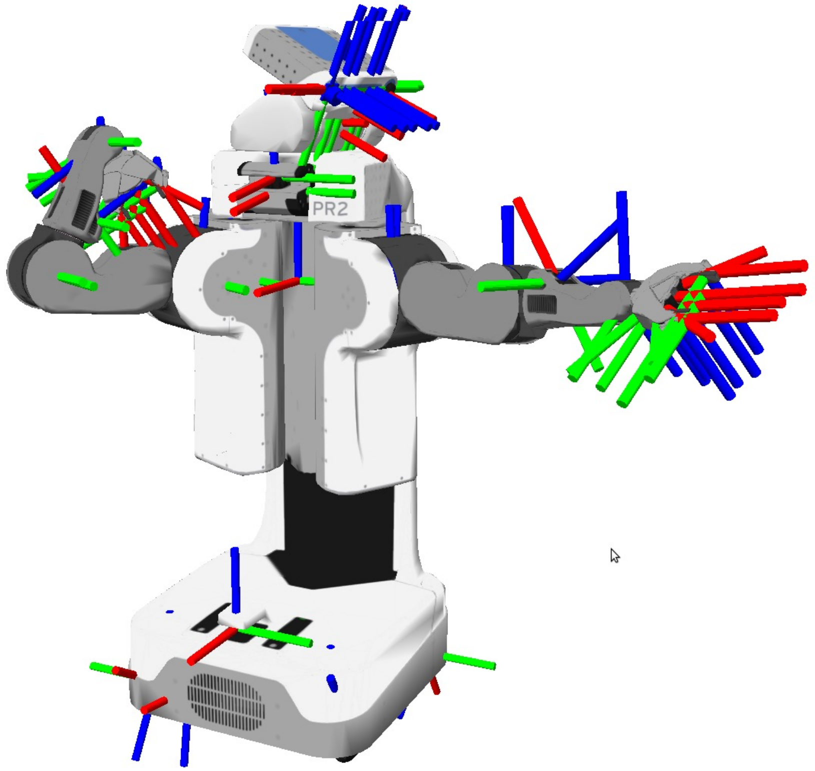
https://alliance.seas.upenn.edu/~meam620/wiki/index.php?n=IanMcMahon2011.Final

Recall from CR04-03

ROS uses a different convention

"If the end effector is a grasping device, the frame should be located at the recommended object grasping location. The frame orientation is defined as X the axis going 'toward' the object. Y the main dimension in which the grasping device moves and Z orthogonal to X and Y axes."

https://www.ros.org/reps/rep-0120.html#l-gripper-and-r-gripper



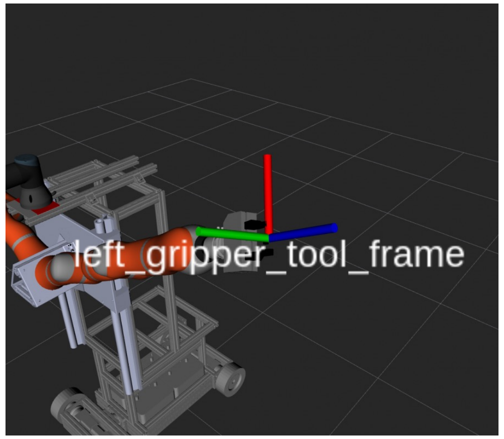
http://library.isr.ist.utl.pt/docs/roswiki/tf2.html

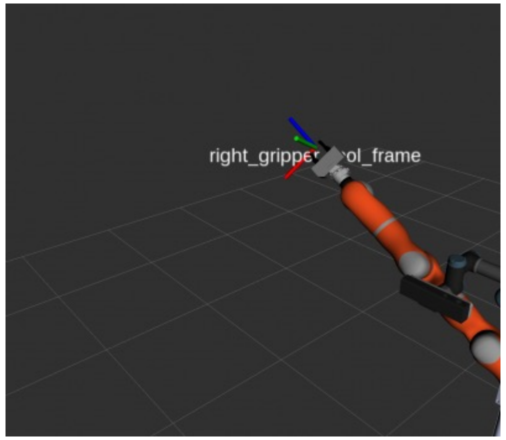
Recall from CR04-03

CRAM uses third convention

The frame orientation is defined as X the main dimension in which the grasping device moves, Y orthogonal to X and Z axes, and Z the axis going "toward" the object

This is similar to the standard approach, but with a rotation of 90° about the Z axis





http://cram-system.org/tutorials/demo/fetch_and_place

Defining a New Grasp Pose

Thus, to make the front-left diagonal grasp

- The Z axis of the gripper should be aligned at 45 degrees in between the X and Y axes of the object
- The X axis should be perpendicular to the Z axis of the bottle (since we are grasping across the Z axis of the bottle)
- The Y of the gripper aligned with the Z axis of the bottle

If we were using the standard convention, it would be the Y axis that is perpendicular to the Z axis of the bottle, since the Y axis would be aligned with the direction of motion of the gripper, not the X axis as it is in CRAM

Defining a New Grasp Pose

To achieve this orientation, we apply the following rotations to the identity pose

- First, rotate 90° about the X axis

If we were using the standard convention, we would rotation about the Y axis, since the Y axis would be aligned with the direction of motion of the gripper, not the X axis as it is in CRAM

- Then rotate -45° about the Y axis of the new (station) frame

- Recall from CR04-01:

$$
\begin{aligned}
\boldsymbol{H} &= \boldsymbol{Rot}(X, {}^{\pi}/_{2})\boldsymbol{Rot}(Y, -{}^{\pi}/_{4}) \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos {}^{\pi}/_{2} & -\sin {}^{\pi}/_{2} & 0 \\ 0 & \sin {}^{\pi}/_{2} & \cos {}^{\pi}/_{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos -{}^{\pi}/_{4} & 0 & \sin -{}^{\pi}/_{4} & 0 \\ 0 & 1 & 0 & 0 \\ -\sin -{}^{\pi}/_{4} & 0 & \cos -{}^{\pi}/_{4} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos -{}^{\pi}/_{4} & 0 & \sin -{}^{\pi}/_{4} & 0 \\ 0 & 1 & 0 & 0 \\ -\sin -{}^{\pi}/_{4} & 0 & \cos -{}^{\pi}/_{4} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos -{}^{\pi}/_{4} & 0 & \sin -{}^{\pi}/_{4} & 0 \\ \sin -{}^{\pi}/_{4} & 0 & -\cos -{}^{\pi}/_{4} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \sin {}^{\pi}/_{4} & 0 & -\sin {}^{\pi}/_{4} & 0 \\ -\sin {}^{\pi}/_{4} & 0 & -\sin {}^{\pi}/_{4} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

```
(defparameter *sin-pi/4* (sin (/ pi 4)))
(defparameter *-sin-pi/4* (- (sin (/ pi 4))))

(defparameter *diagonal-rotation*
        `((,*sin-pi/4* 0 ,*-sin-pi/4*)
          (,*-sin-pi/4* 0 ,*-sin-pi/4*)
          (0 1 0)))
```

Since sin(pi/4) = cos(pi/4),
we have only defined one variable
and used it interchangeably in the
defined rotation matrix

Note the backquote
and the comma operators
to evaluate the elements of this list

we have defined a grasp called fl-diagonal
for objects drink and bottle,
which can be accessed with the left or right arm of the robot

```
(cram-object-interfaces:def-object-type-to-gripper-transforms '(:drink :bottle) '(:left :right) :fl-diagonal
  :grasp-translation `(,(- *bottle-grasp-xy-offset*) ,(- *bottle-grasp-xy-offset*) ,*bottle-grasp-z-offset*)
  :grasp-rot-matrix *diagonal-rotation*
  :pregrasp-offsets `(,*bottle-pregrasp-xy-offset* ,*bottle-pregrasp-xy-offset* ,*lift-z-offset*)
  :lift-offsets *lift-offset*)
```

we have defined a grasp called fl-diagonal
for objects drink and bottle,
which can be accessed with the left or right arm of the robot

```
(cram-object-interfaces:def-object-type-to-gripper-transforms '(:drink :bottle) '(:left :right) :fl-diagonal
  :grasp-translation `(,(- *bottle-grasp-xy-offset*) ,(- *bottle-grasp-xy-offset*) ,*bottle-grasp-z-offset*)
  :grasp-rot-matrix *diagonal-rotation*
  :pregrasp-offsets `(,*bottle-pregrasp-xy-offset* ,*bottle-pregrasp-xy-offset* ,*lift-z-offset*)
  :lift-offsets *lift-offset*)
```

we have defined a grasp called fl-diagonal
for objects drink and bottle,
which can be accessed with the left or right arm of the robot

```
(cram-object-interfaces:def-object-type-to-gripper-transforms '(:drink :bottle) '(:left :right) :fl-diagonal
  :grasp-translation `(,(- *bottle-grasp-xy-offset*) ,(- *bottle-grasp-xy-offset*) ,*bottle-grasp-z-offset*)
  :grasp-rot-matrix *diagonal-rotation*
  :pregrasp-offsets `(,*bottle-pregrasp-xy-offset* ,*bottle-pregrasp-xy-offset* ,*lift-z-offset*)
  :lift-offsets *lift-offset*)
```

```
(cram-object-interfaces:def-object-type-to-gripper-transforms '(:drink :bottle) '(:left :right) :fl-diagonal
  :grasp-translation `(,(- *bottle-grasp-xy-offset*) ,(- *bottle-grasp-xy-offset*) ,*bottle-grasp-z-offset*)
  :grasp-rot-matrix *diagonal-rotation*
  :pregrasp-offsets `(,*bottle-pregrasp-xy-offset* ,*bottle-pregrasp-xy-offset* ,*lift-z-offset*)
  :lift-offsets *lift-offset*)
```

The grasp-translation gives the position of the gripper (the tool centre position TCP) when grasping,
i.e. the origin of the grasp pose

```
(defparameter *lift-z-offset* 0.15 "in meters")
(defparameter *lift-offset* `(0.0 0.0 ,*lift-z-offset*))

(defparameter *bottle-pregrasp-xy-offset* 0.15 "in meters")
(defparameter *bottle-grasp-xy-offset* 0.02 "in meters")
(defparameter *bottle-grasp-z-offset* 0.005 "in meters")
```

```
(cram-object-interfaces:def-object-type-to-gripper-transforms '(:drink :bottle) '(:left :right) :fl-diagonal
  :grasp-translation `(,(- *bottle-grasp-xy-offset*) ,(- *bottle-grasp-xy-offset*) ,*bottle-grasp-z-offset*)
  :grasp-rot-matrix *diagonal-rotation*
  :pregrasp-offsets `(,*bottle-pregrasp-xy-offset* ,*bottle-pregrasp-xy-offset* ,*lift-z-offset*)
  :lift-offsets *lift-offset*)
```

The grasp-rot-matrix gives the orientation required for grasping, i.e. the orientation of the grasp pose

```
(defparameter *sin-pi/4* (sin (/ pi 4)))
(defparameter *-sin-pi/4* (- (sin (/ pi 4))))

(defparameter *diagonal-rotation*
        `((,*sin-pi/4* 0 ,*-sin-pi/4*)
          (,*-sin-pi/4* 0 ,*-sin-pi/4*)
          (0 1 0)))
```

```lisp
(cram-object-interfaces:def-object-type-to-gripper-transforms '(:drink :bottle) '(:left :right) :fl-diagonal
  :grasp-translation `(,(- *bottle-grasp-xy-offset*) ,(- *bottle-grasp-xy-offset*) ,*bottle-grasp-z-offset*)
  :grasp-rot-matrix *diagonal-rotation*
  :pregrasp-offsets `(,*bottle-pregrasp-xy-offset* ,*bottle-pregrasp-xy-offset* ,*lift-z-offset*)
  :lift-offsets *lift-offset*)
```

The pre-grasp offsets gives you the distance the gripper will be positioned before the grasp
i.e. the origin of the pre-grasp pose

The orientation of the pre-grasp pose is aligned with the grasp pose

```lisp
(defparameter *lift-z-offset* 0.15 "in meters")
(defparameter *lift-offset* `(0.0 0.0 ,*lift-z-offset*))

(defparameter *bottle-pregrasp-xy-offset* 0.15 "in meters")
(defparameter *bottle-grasp-xy-offset* 0.02 "in meters")
(defparameter *bottle-grasp-z-offset* 0.005 "in meters")
```

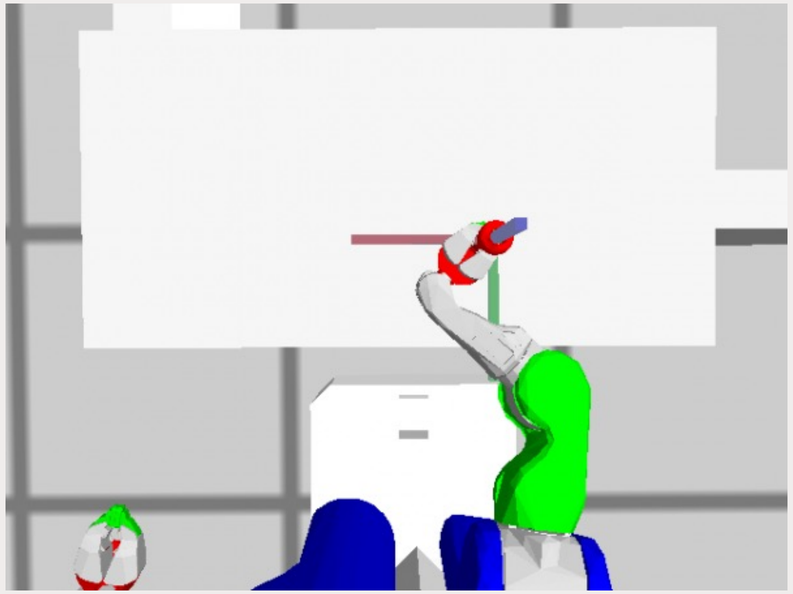
```
(cram-object-interfaces:def-object-type-to-gripper-transforms '(:drink :bottle) '(:left :right) :fl-diagonal
  :grasp-translation `(,(- *bottle-grasp-xy-offset*) ,(- *bottle-grasp-xy-offset*) ,*bottle-grasp-z-offset*)
  :grasp-rot-matrix *diagonal-rotation*
  :pregrasp-offsets `(,*bottle-pregrasp-xy-offset* ,*bottle-pregrasp-xy-offset* ,*lift-z-offset*)
  :lift-offsets *lift-offset*)
```

The lift-offsets gives you the distance the gripper will be positioned after the grasp
i.e. the origin of the lift pose

```
(defparameter *lift-z-offset* 0.15 "in meters")
(defparameter *lift-offset* `(0.0 0.0 ,*lift-z-offset*))

(defparameter *bottle-pregrasp-xy-offset* 0.15 "in meters")
(defparameter *bottle-grasp-xy-offset* 0.02 "in meters")
(defparameter *bottle-grasp-z-offset* 0.005 "in meters")
```

```
(spawn-object '((-1.6 -0.9 0.82) (0 0 0 1)))

(pr2-proj:with-simulated-robot
  (let ((?navigation-goal *base-pose-near-table*))
    (cpl:par
      (exe:perform (desig:a motion
                            (type moving-torso)
                            (joint-angle 0.3)))
      (park-arms)
      ;; Moving the robot near the table.
      (exe:perform (desig:a motion
                            (type going)
                            (target (desig:a location
                                             (pose ?navigation-goal)))))))
  ;; Looking towards the bottle before perceiving.
  (let ((?looking-direction *downward-look-coordinate*))
    (exe:perform (desig:a motion
                          (type looking)
                          (target (desig:a location
                                           (pose ?looking-direction))))))
  ;; Detect the bottle on the table.
  (setf *perceived-bottle* (exe:perform (desig:a motion
                                                 (type detecting)
                                                 (object (desig:an object
                                                                   (type :bottle))))))

  (let ((?perceived-bottle *perceived-bottle*))
    (exe:perform (desig:an action
                           (type picking-up)
                           (arm right)
                           (grasp fl-diagonal)
                           (object ?perceived-bottle)))))
```

pick up with a front-left diagonal grasp

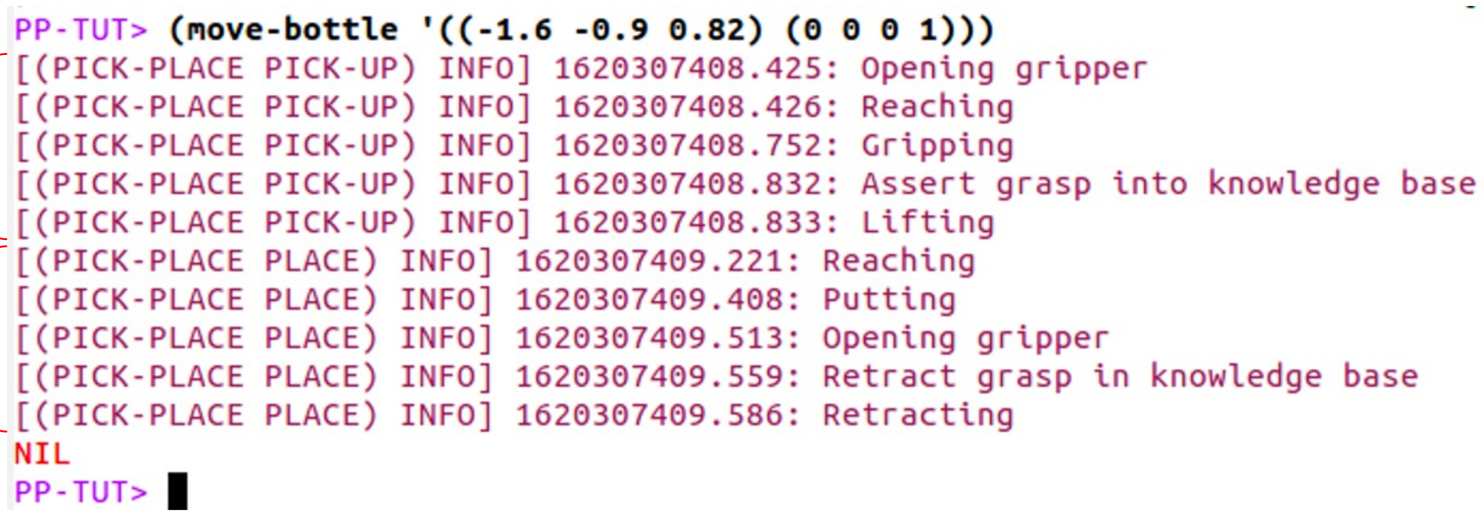


```
(visualize-coordinates (btr:link-pose (btr:get-robot-object) "r_gripper_tool_frame"))
```

Simple Fetch and Place Plan

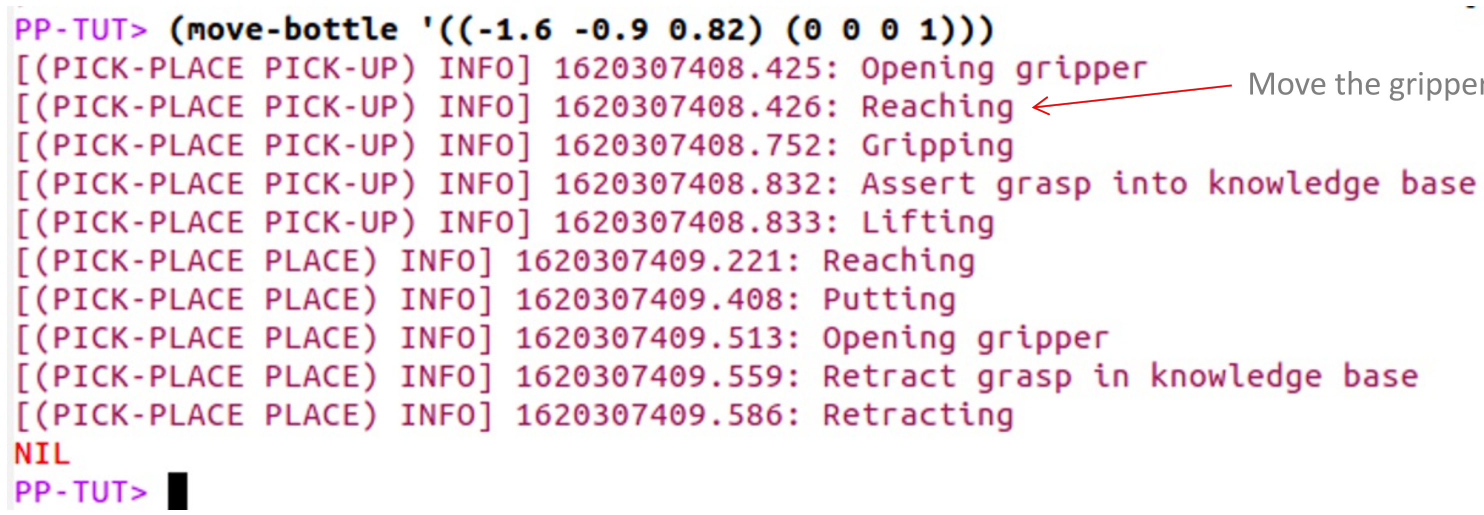
The pick-up action designator is resolved into four atomic action designators

The place action designator is resolved into four atomic action designators



```
PP-TUT> (move-bottle '((-1.6 -0.9 0.82) (0 0 0 1)))
[(PICK-PLACE PICK-UP) INFO] 1620307408.425: Opening gripper
[(PICK-PLACE PICK-UP) INFO] 1620307408.426: Reaching
[(PICK-PLACE PICK-UP) INFO] 1620307408.752: Gripping
[(PICK-PLACE PICK-UP) INFO] 1620307408.832: Assert grasp into knowledge base
[(PICK-PLACE PICK-UP) INFO] 1620307408.833: Lifting
[(PICK-PLACE PLACE) INFO] 1620307409.221: Reaching
[(PICK-PLACE PLACE) INFO] 1620307409.408: Putting
[(PICK-PLACE PLACE) INFO] 1620307409.513: Opening gripper
[(PICK-PLACE PLACE) INFO] 1620307409.559: Retract grasp in knowledge base
[(PICK-PLACE PLACE) INFO] 1620307409.586: Retracting
NIL
PP-TUT> █
```

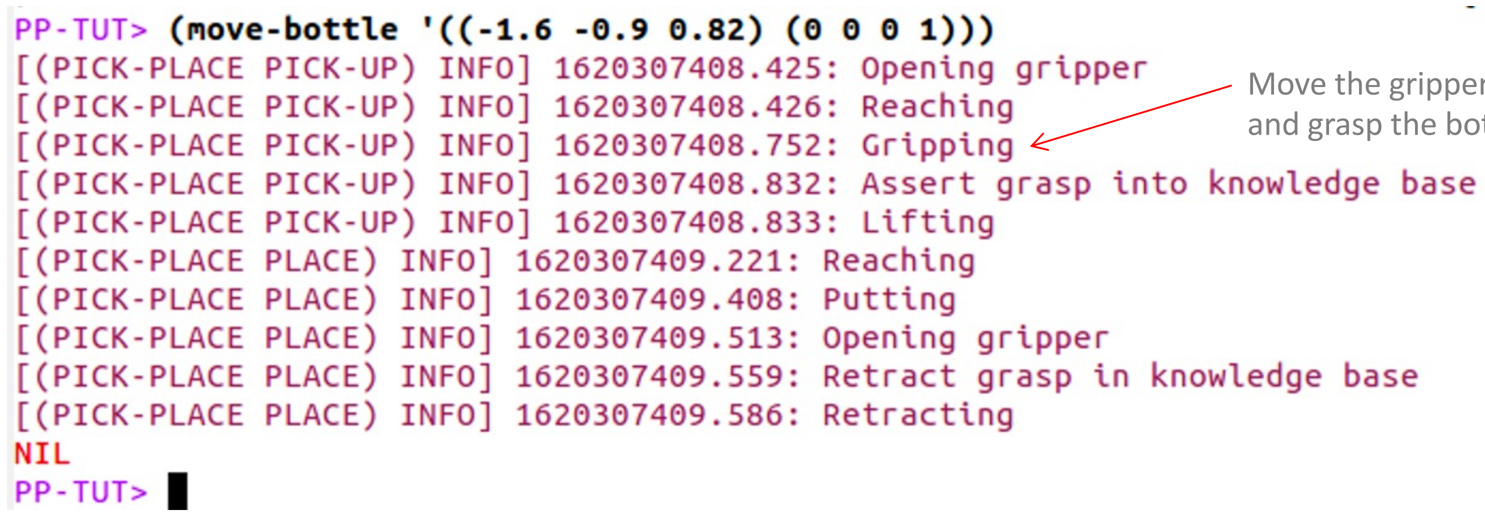
Simple Fetch and Place Plan



```
PP-TUT> (move-bottle '((-1.6 -0.9 0.82) (0 0 0 1)))
[(PICK-PLACE PICK-UP) INFO] 1620307408.425: Opening gripper
[(PICK-PLACE PICK-UP) INFO] 1620307408.426: Reaching
[(PICK-PLACE PICK-UP) INFO] 1620307408.752: Gripping
[(PICK-PLACE PICK-UP) INFO] 1620307408.832: Assert grasp into knowledge base
[(PICK-PLACE PICK-UP) INFO] 1620307408.833: Lifting
[(PICK-PLACE PLACE) INFO] 1620307409.221: Reaching
[(PICK-PLACE PLACE) INFO] 1620307409.408: Putting
[(PICK-PLACE PLACE) INFO] 1620307409.513: Opening gripper
[(PICK-PLACE PLACE) INFO] 1620307409.559: Retract grasp in knowledge base
[(PICK-PLACE PLACE) INFO] 1620307409.586: Retracting
NIL
PP-TUT> 
```

Move the gripper to the pre-grasp pose

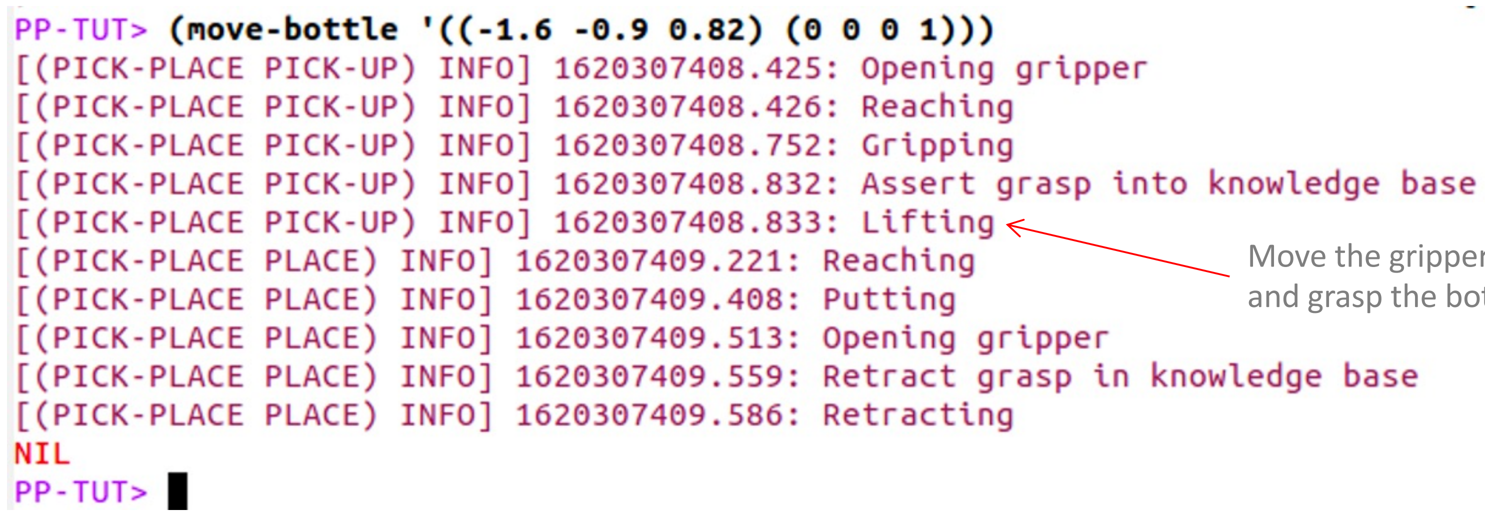
Simple Fetch and Place Plan



```
PP-TUT> (move-bottle '((-1.6 -0.9 0.82) (0 0 0 1)))
[(PICK-PLACE PICK-UP) INFO] 1620307408.425: Opening gripper
[(PICK-PLACE PICK-UP) INFO] 1620307408.426: Reaching
[(PICK-PLACE PICK-UP) INFO] 1620307408.752: Gripping
[(PICK-PLACE PICK-UP) INFO] 1620307408.832: Assert grasp into knowledge base
[(PICK-PLACE PICK-UP) INFO] 1620307408.833: Lifting
[(PICK-PLACE PLACE) INFO] 1620307409.221: Reaching
[(PICK-PLACE PLACE) INFO] 1620307409.408: Putting
[(PICK-PLACE PLACE) INFO] 1620307409.513: Opening gripper
[(PICK-PLACE PLACE) INFO] 1620307409.559: Retract grasp in knowledge base
[(PICK-PLACE PLACE) INFO] 1620307409.586: Retracting
NIL
PP-TUT>
```

Move the gripper to the grasp pose and grasp the bottle

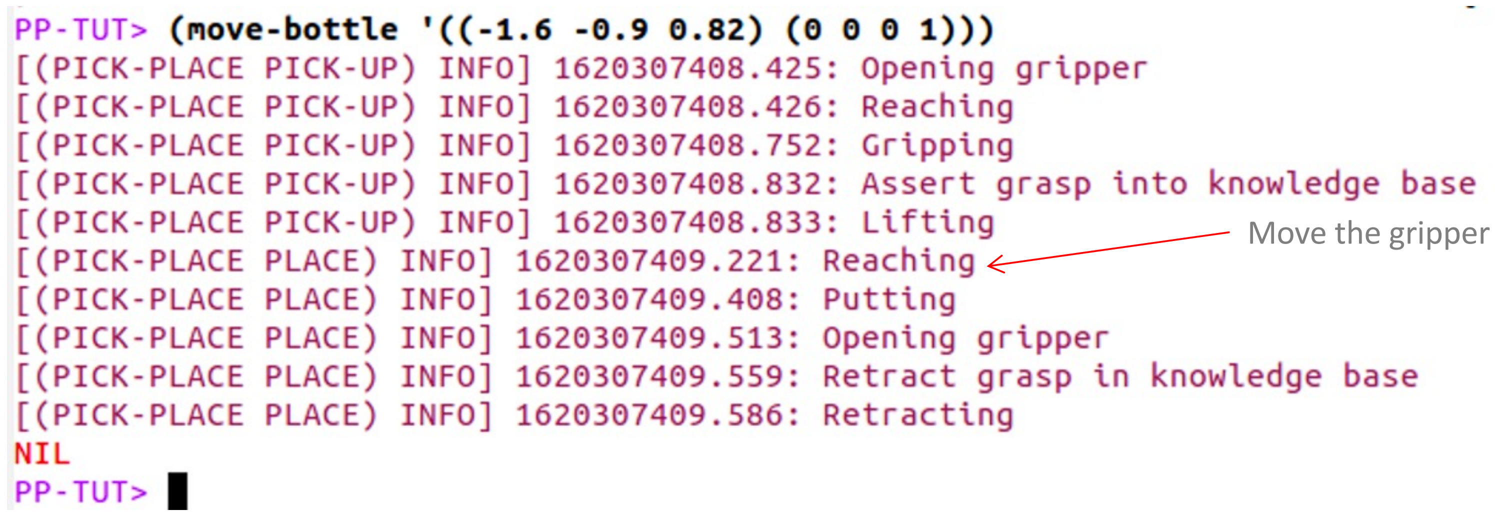
Simple Fetch and Place Plan



```
PP-TUT> (move-bottle '((-1.6 -0.9 0.82) (0 0 0 1)))
[(PICK-PLACE PICK-UP) INFO] 1620307408.425: Opening gripper
[(PICK-PLACE PICK-UP) INFO] 1620307408.426: Reaching
[(PICK-PLACE PICK-UP) INFO] 1620307408.752: Gripping
[(PICK-PLACE PICK-UP) INFO] 1620307408.832: Assert grasp into knowledge base
[(PICK-PLACE PICK-UP) INFO] 1620307408.833: Lifting
[(PICK-PLACE PLACE) INFO] 1620307409.221: Reaching
[(PICK-PLACE PLACE) INFO] 1620307409.408: Putting
[(PICK-PLACE PLACE) INFO] 1620307409.513: Opening gripper
[(PICK-PLACE PLACE) INFO] 1620307409.559: Retract grasp in knowledge base
[(PICK-PLACE PLACE) INFO] 1620307409.586: Retracting
NIL
PP-TUT> 
```

Move the gripper to the lift pose
and grasp the bottle

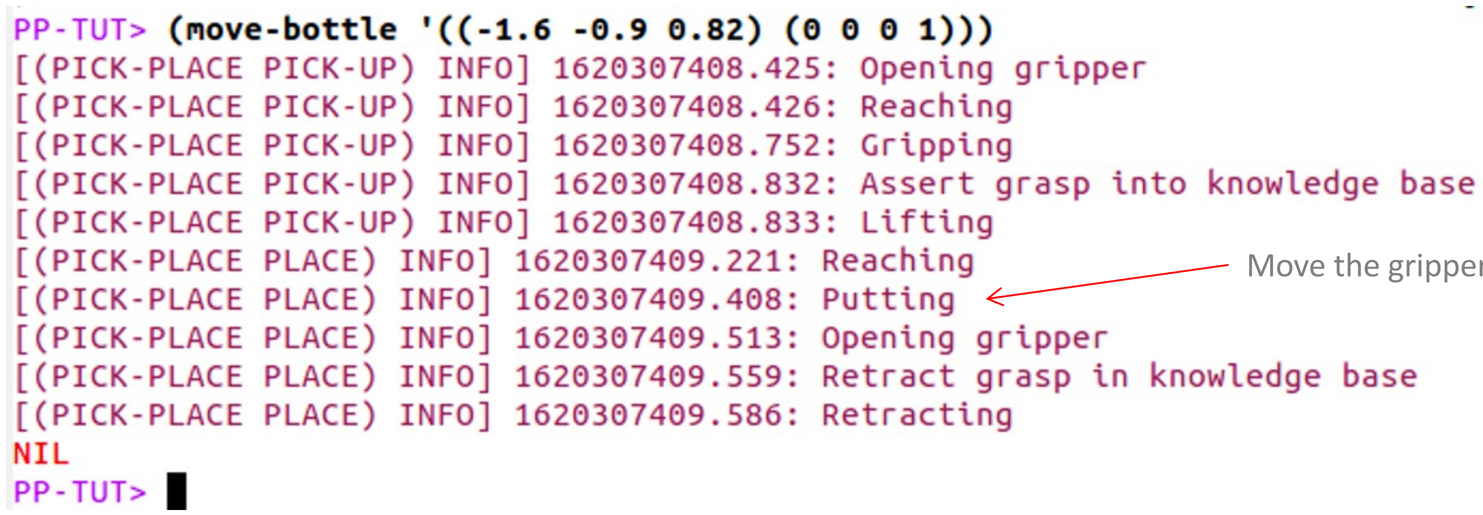
Simple Fetch and Place Plan



```
PP-TUT> (move-bottle '((-1.6 -0.9 0.82) (0 0 0 1)))
[(PICK-PLACE PICK-UP) INFO] 1620307408.425: Opening gripper
[(PICK-PLACE PICK-UP) INFO] 1620307408.426: Reaching
[(PICK-PLACE PICK-UP) INFO] 1620307408.752: Gripping
[(PICK-PLACE PICK-UP) INFO] 1620307408.832: Assert grasp into knowledge base
[(PICK-PLACE PICK-UP) INFO] 1620307408.833: Lifting
[(PICK-PLACE PLACE) INFO] 1620307409.221: Reaching
[(PICK-PLACE PLACE) INFO] 1620307409.408: Putting
[(PICK-PLACE PLACE) INFO] 1620307409.513: Opening gripper
[(PICK-PLACE PLACE) INFO] 1620307409.559: Retract grasp in knowledge base
[(PICK-PLACE PLACE) INFO] 1620307409.586: Retracting
NIL
PP-TUT> 
```

Move the gripper to the lift pose

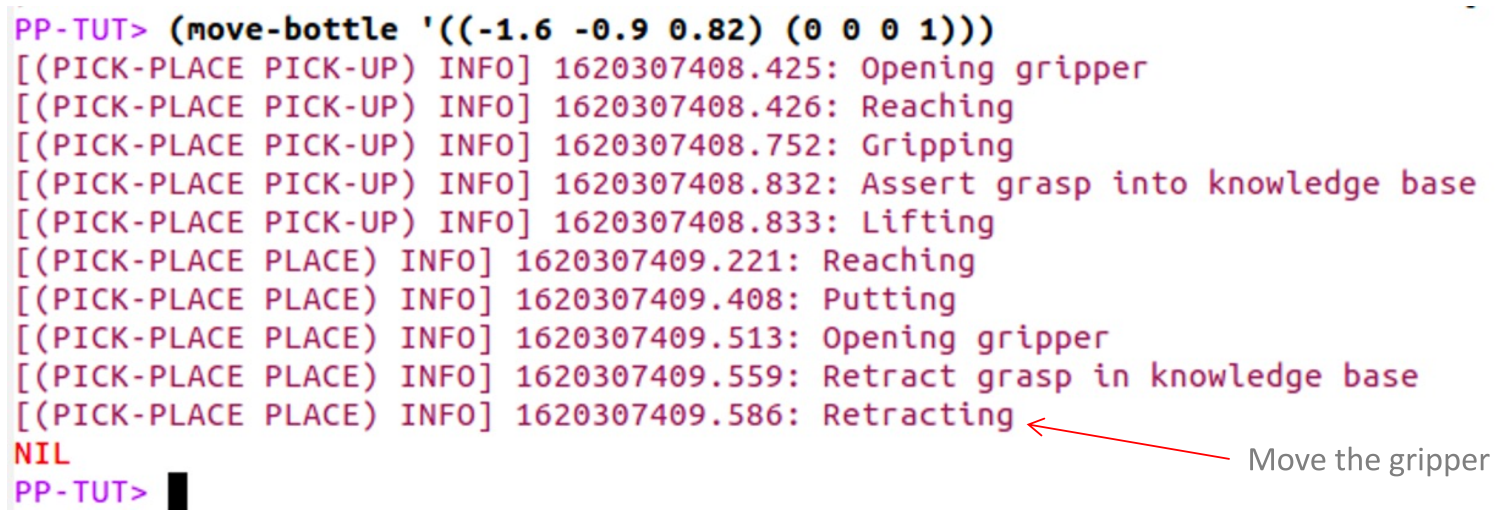
Simple Fetch and Place Plan



```
PP-TUT> (move-bottle '((-1.6 -0.9 0.82) (0 0 0 1)))
[(PICK-PLACE PICK-UP) INFO] 1620307408.425: Opening gripper
[(PICK-PLACE PICK-UP) INFO] 1620307408.426: Reaching
[(PICK-PLACE PICK-UP) INFO] 1620307408.752: Gripping
[(PICK-PLACE PICK-UP) INFO] 1620307408.832: Assert grasp into knowledge base
[(PICK-PLACE PICK-UP) INFO] 1620307408.833: Lifting
[(PICK-PLACE PLACE) INFO] 1620307409.221: Reaching
[(PICK-PLACE PLACE) INFO] 1620307409.408: Putting
[(PICK-PLACE PLACE) INFO] 1620307409.513: Opening gripper
[(PICK-PLACE PLACE) INFO] 1620307409.559: Retract grasp in knowledge base
[(PICK-PLACE PLACE) INFO] 1620307409.586: Retracting
NIL
PP-TUT>
```

Move the gripper to the grasp pose

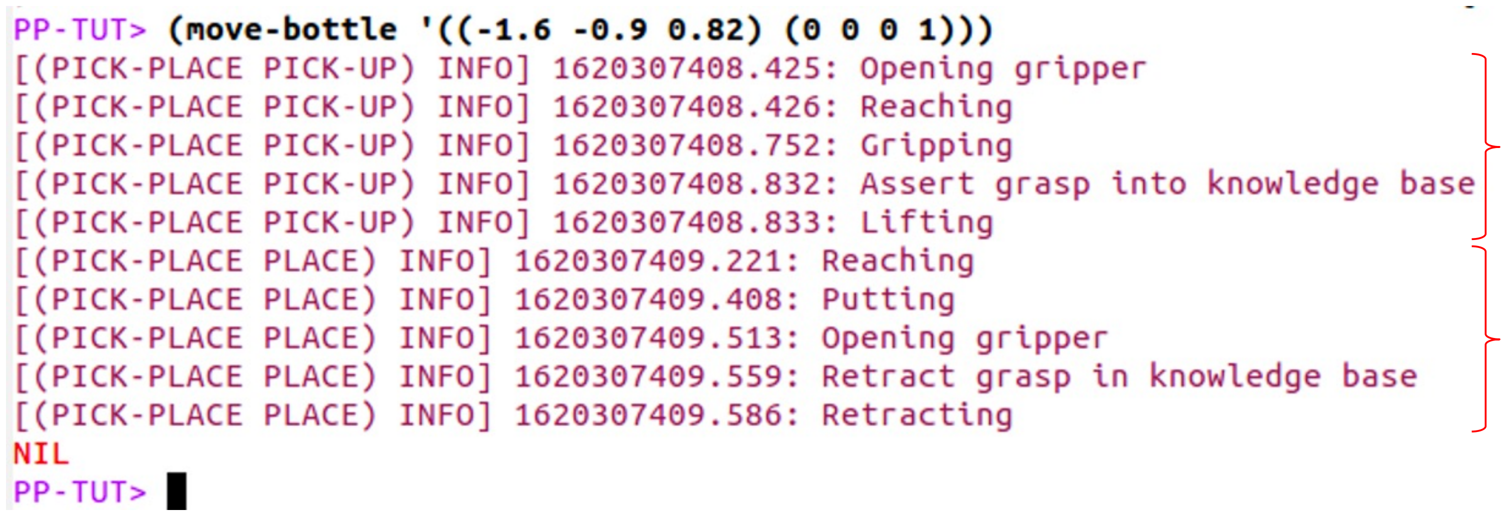
Simple Fetch and Place Plan



```
PP-TUT> (move-bottle '((-1.6 -0.9 0.82) (0 0 0 1)))
[(PICK-PLACE PICK-UP) INFO] 1620307408.425: Opening gripper
[(PICK-PLACE PICK-UP) INFO] 1620307408.426: Reaching
[(PICK-PLACE PICK-UP) INFO] 1620307408.752: Gripping
[(PICK-PLACE PICK-UP) INFO] 1620307408.832: Assert grasp into knowledge base
[(PICK-PLACE PICK-UP) INFO] 1620307408.833: Lifting
[(PICK-PLACE PLACE) INFO] 1620307409.221: Reaching
[(PICK-PLACE PLACE) INFO] 1620307409.408: Putting
[(PICK-PLACE PLACE) INFO] 1620307409.513: Opening gripper
[(PICK-PLACE PLACE) INFO] 1620307409.559: Retract grasp in knowledge base
[(PICK-PLACE PLACE) INFO] 1620307409.586: Retracting
NIL
PP-TUT> 
```

Move the gripper to the pre-grasp pose

Simple Fetch and Place Plan


```
PP-TUT> (move-bottle '((-1.6 -0.9 0.82) (0 0 0 1)))
[(PICK-PLACE PICK-UP) INFO] 1620307408.425: Opening gripper
[(PICK-PLACE PICK-UP) INFO] 1620307408.426: Reaching
[(PICK-PLACE PICK-UP) INFO] 1620307408.752: Gripping
[(PICK-PLACE PICK-UP) INFO] 1620307408.832: Assert grasp into knowledge base
[(PICK-PLACE PICK-UP) INFO] 1620307408.833: Lifting
[(PICK-PLACE PLACE) INFO] 1620307409.221: Reaching
[(PICK-PLACE PLACE) INFO] 1620307409.408: Putting
[(PICK-PLACE PLACE) INFO] 1620307409.513: Opening gripper
[(PICK-PLACE PLACE) INFO] 1620307409.559: Retract grasp in knowledge base
[(PICK-PLACE PLACE) INFO] 1620307409.586: Retracting
NIL
PP-TUT> █
```

Note the order
pre-grasp pose
grasp pose
lift pose

Note the order
lift pose
grasp pose
pre-grasp pose

Recommended Reading

CRAM zero prerequisites demo tutorial: simple fetch and place

`http://cram-system.org/tutorials/demo/fetch_and_place`

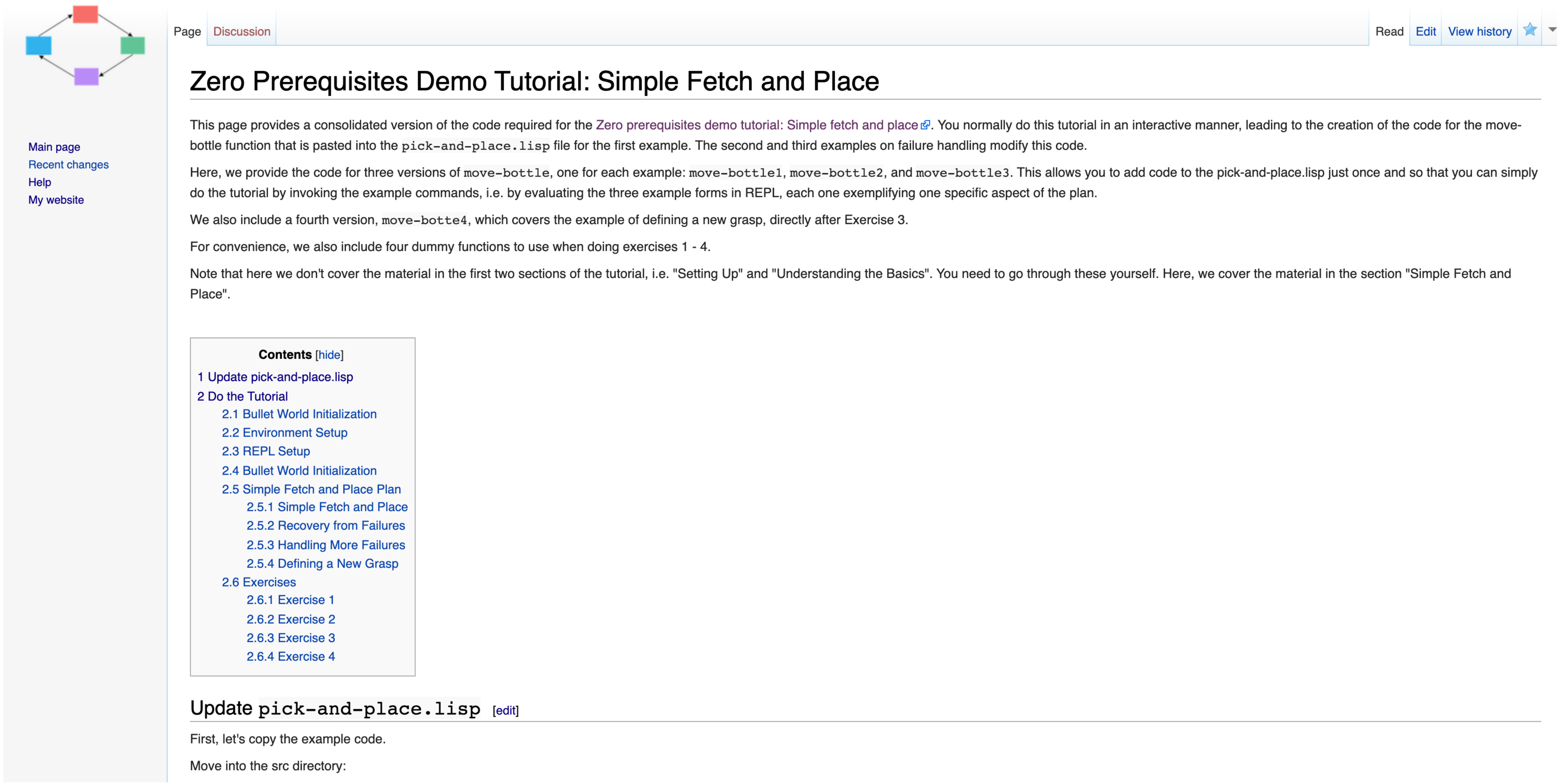
Implementation of a pick-and-place CRAM plan

Follow these instructions

"Zero Prerequisites Demo Tutorial: Simple Fetch and Place"

http://www.vernon.eu/wiki/Zero_Prerequisites_Demo_Tutorial:_Simple_Fetch_and_Place

to implement the pick-and-place example


Page Discussion

Read Edit View history

Main page
Recent changes
Help
My website

Zero Prerequisites Demo Tutorial: Simple Fetch and Place

This page provides a consolidated version of the code required for the Zero prerequisites demo tutorial: Simple fetch and place. You normally do this tutorial in an interactive manner, leading to the creation of the code for the move-bottle function that is pasted into the `pick-and-place.lisp` file for the first example. The second and third examples on failure handling modify this code.

Here, we provide the code for three versions of `move-bottle`, one for each example: `move-bottle1`, `move-bottle2`, and `move-bottle3`. This allows you to add code to the pick-and-place.lisp just once and so that you can simply do the tutorial by invoking the example commands, i.e. by evaluating the three example forms in REPL, each one exemplifying one specific aspect of the plan.

We also include a fourth version, `move-botte4`, which covers the example of defining a new grasp, directly after Exercise 3.

For convenience, we also include four dummy functions to use when doing exercises 1 - 4.

Note that here we don't cover the material in the first two sections of the tutorial, i.e. "Setting Up" and "Understanding the Basics". You need to go through these yourself. Here, we cover the material in the section "Simple Fetch and Place".

Contents [hide]

Update `pick-and-place.lisp` [edit]

First, let's copy the example code.

Move into the src directory:

http://www.vernon.eu/wiki/Zero_Prerequisites_Demo_Tutorial:_Simple_Fetch_and_Place