Hybrid Reasoning for Mobile Manipulation based on Object Knowledge

Daniel Leidner and Christoph Borst

Institute of Robotics and Mechatronics, German Aerospace Center (DLR)

DLR

Motivation

Given a manipulation task, a robot has to reason about the *symbolic* order of the actions, but also about the *geometric* execution.

Detailed knowledge about the *functionality* of the objects is required.

Mobile manipulation tasks require additional *spatial knowledge*.

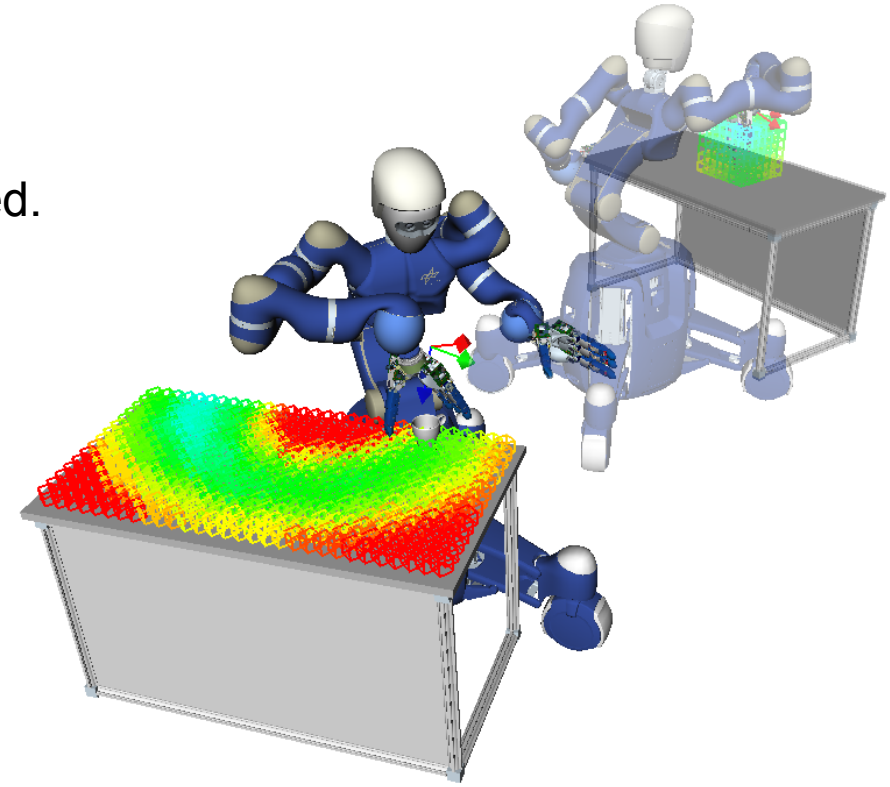
DLR

Motivation

Given a manipulation task, a robot has to reason about the *symbolic* order of the actions, but also about the *geometric* execution.

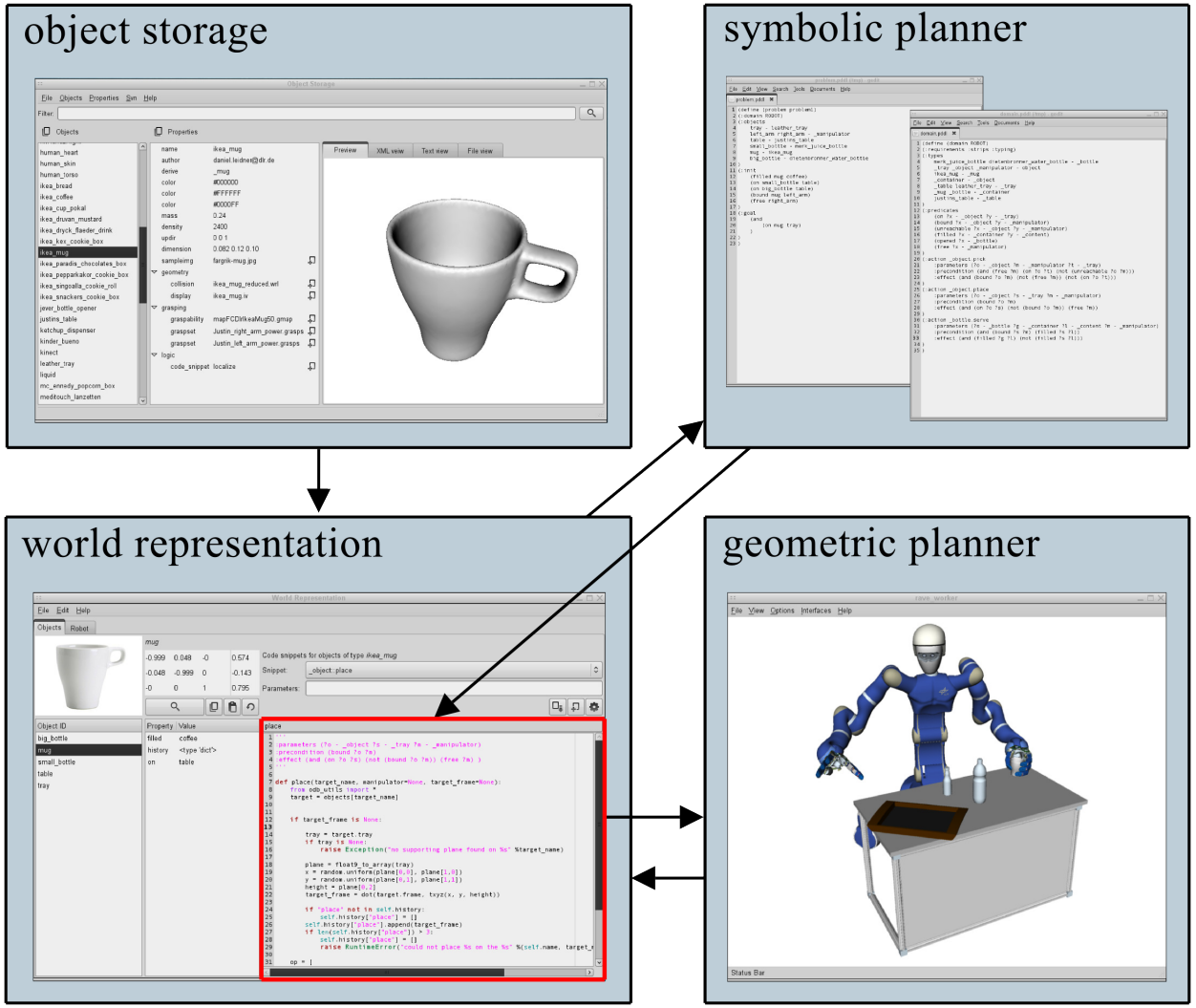
Detailed knowledge about the *functionality* of the objects is required.

Mobile manipulation tasks require additional *spatial knowledge*.



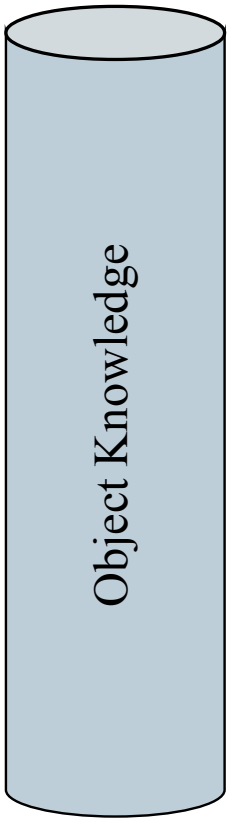
DLR

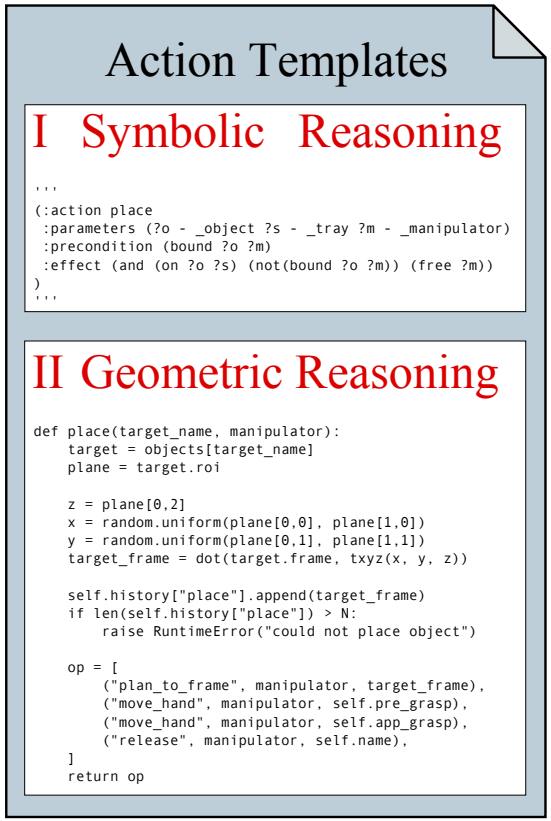
Integrating Object Knowledge into Hybrid Reasoning

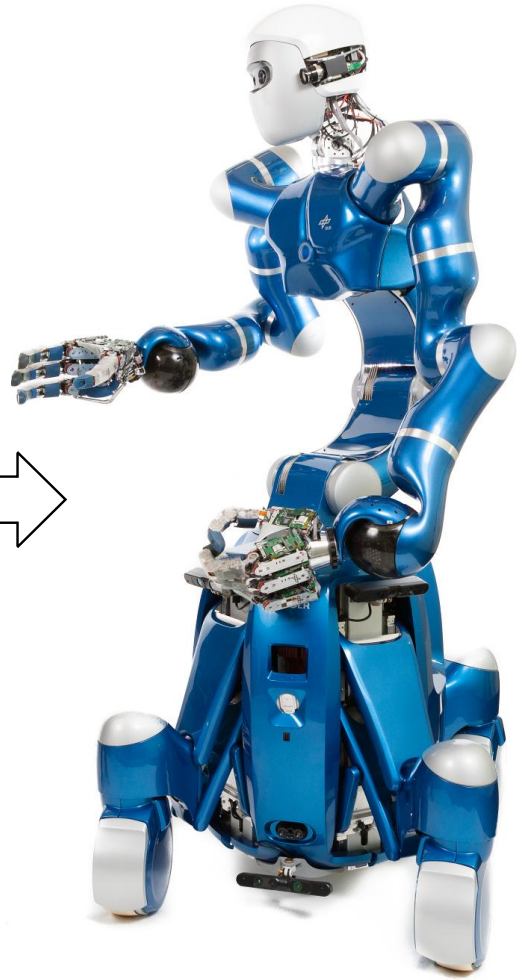


DLR

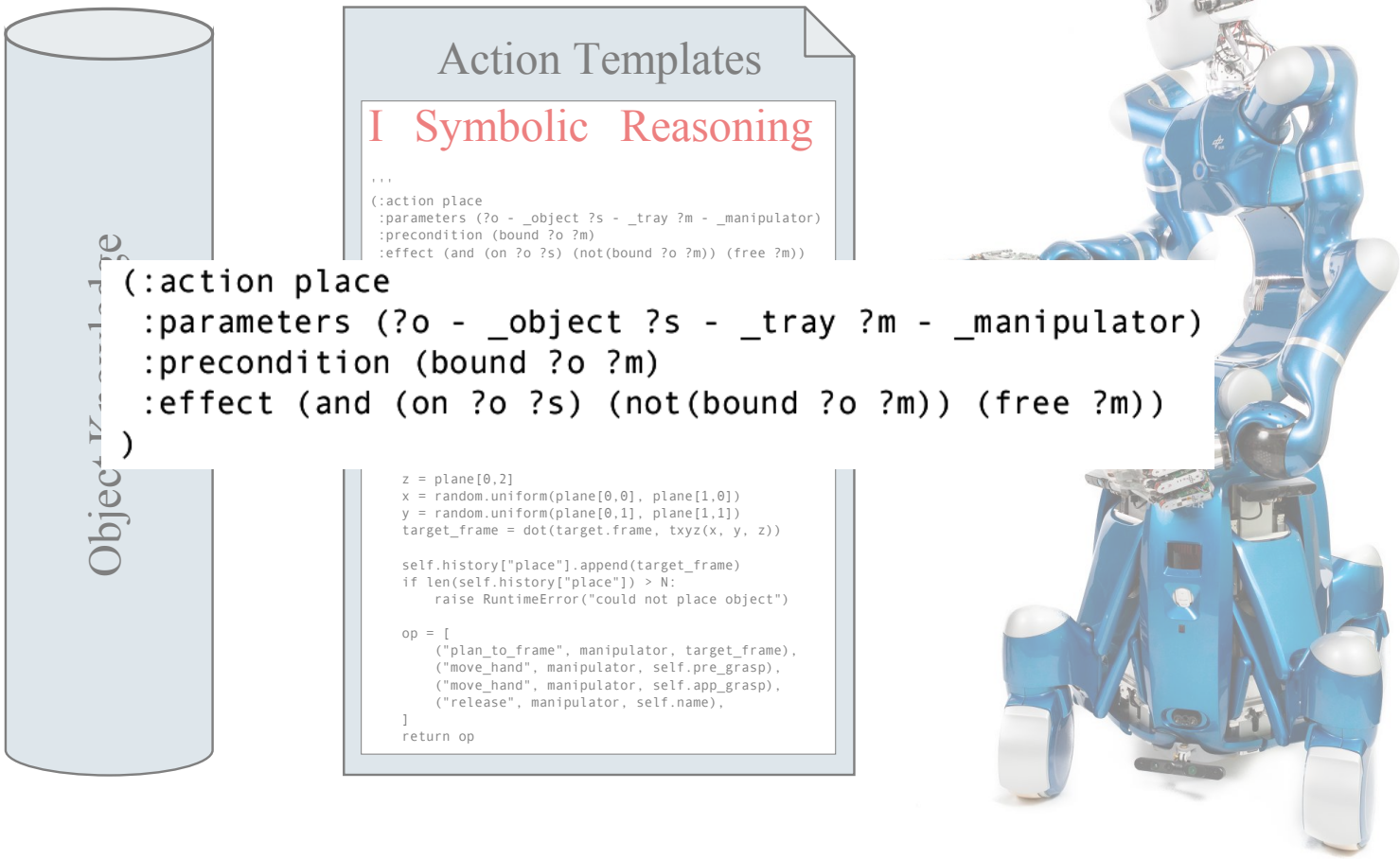
Integrating Object Knowledge into Hybrid Reasoning (cont.)







Integrating Object Knowledge into Hybrid Reasoning (cont.)



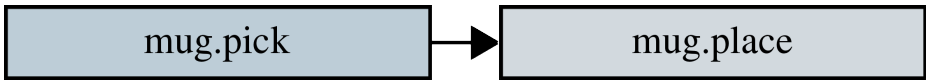
Integrating Object Knowledge into Hybrid Reasoning (cont.)


```
def place(target_name, manipulator):
    target = objects[target_name]
    plane = target.roi

    z = plane[0,2]
    x = random.uniform(plane[0,0], plane[1,0])
    y = random.uniform(plane[0,1], plane[1,1])
    target_frame = dot(target.frame, txyz(x, y, z))

    self.history["place"].append(target_frame)
    if len(self.history["place"]) > N:
        raise RuntimeError("could not place object")

    op = [
        ("plan_to_frame", manipulator, target_frame),
        ("move_hand", manipulator, self.pre_grasp),
        ("move_hand", manipulator, self.app_grasp),
        ("release", manipulator, self.name),
    ]
    return op
```

DLR

Hybrid Reasoning Procedure

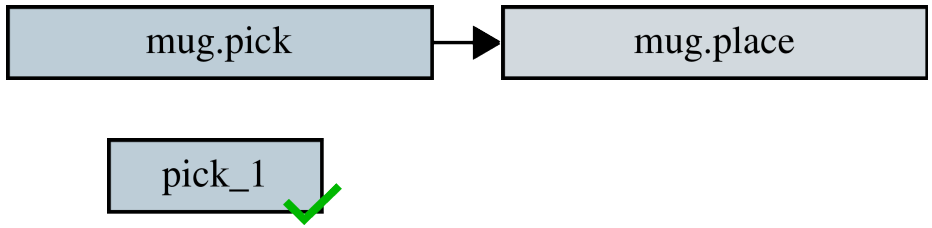
A symbolic transition is provided by a symbolic planner



Hybrid Reasoning Procedure

A symbolic transition is provided by a symbolic planner

Geometric alternatives are attempt

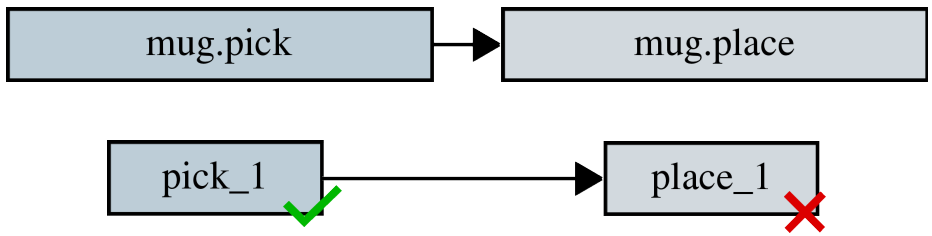


DLR

Hybrid Reasoning Procedure

A symbolic transition is provided by a symbolic planner

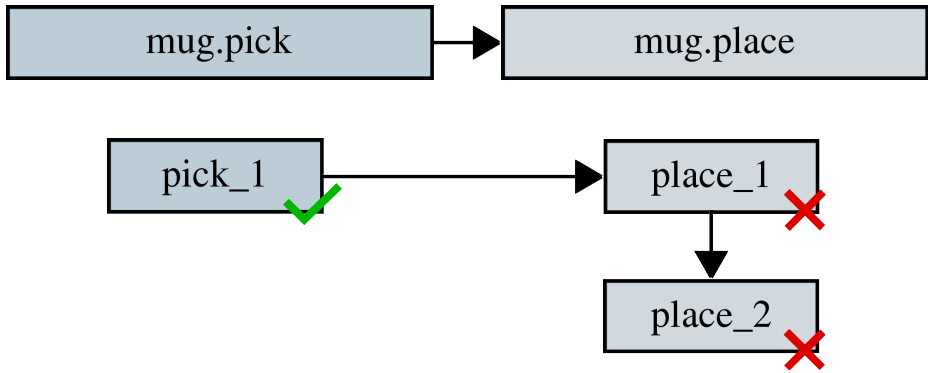
Geometric alternatives are attempt repeatedly in case of failure



Hybrid Reasoning Procedure

A symbolic transition is provided by a symbolic planner

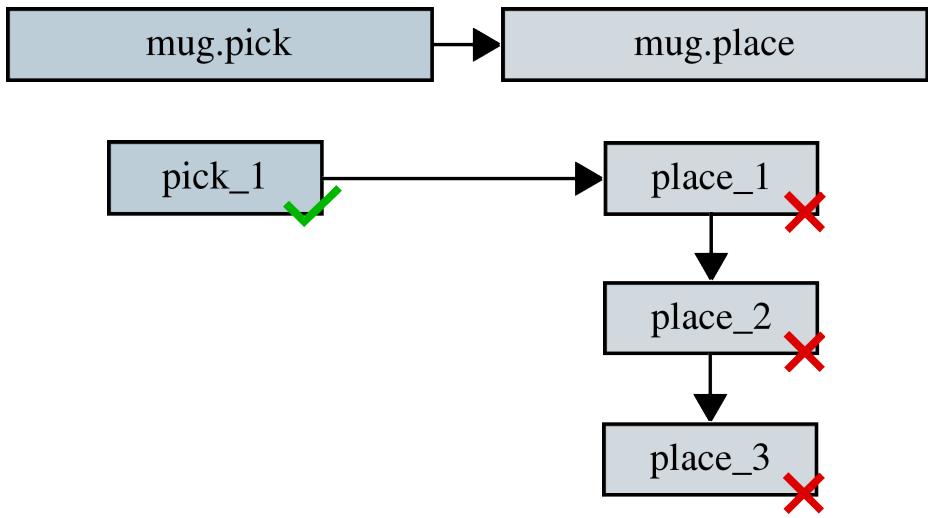
Geometric alternatives are attempt repeatedly in case of failure



Hybrid Reasoning Procedure

A symbolic transition is provided by a symbolic planner

Geometric alternatives are attempt repeatedly in case of failure



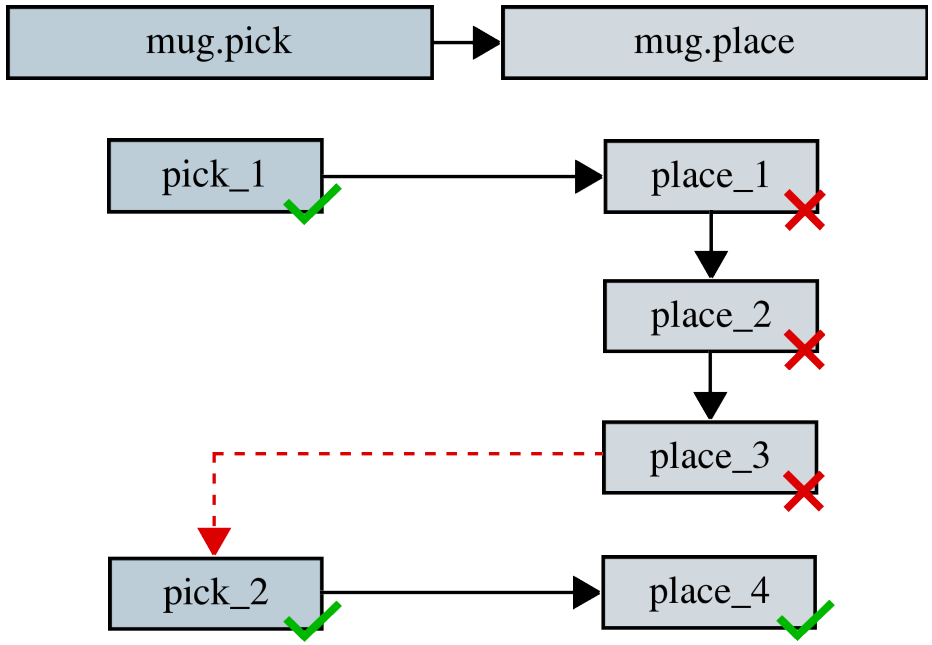
DLR

Hybrid Reasoning Procedure

A symbolic transition is provided by a symbolic planner

Geometric alternatives are attempt repeatedly in case of failure

If all alternatives have been attempted a backtracking mechanism is initiated

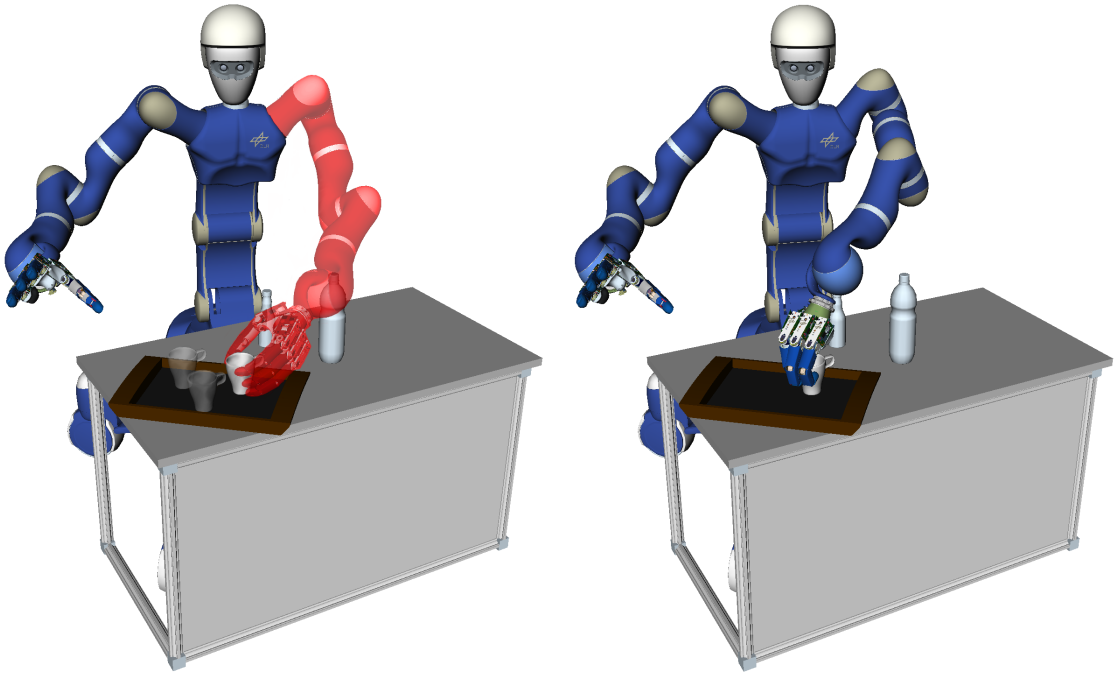


Hybrid Reasoning Procedure

A symbolic transition is provided by a symbolic planner

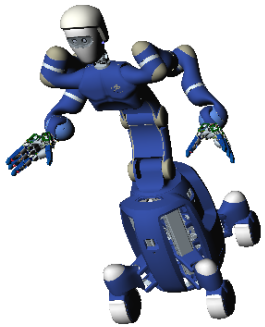
Geometric alternatives are attempt repeatedly in case of failure

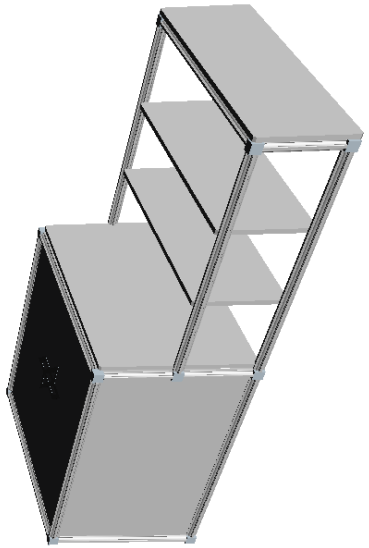
If all alternatives have been attempted a backtracking mechanism is initiated



DLR

Base Positioning for Mobile Manipulation



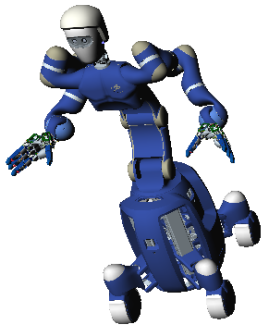


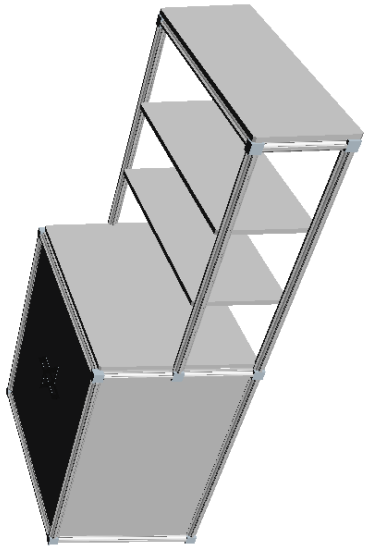
```
(:goal(and (on mug1 shelf) (on mug2 shelf) ))
```

DLR

Base Positioning for Mobile Manipulation

Where should a mobile robot move to fulfill a task?





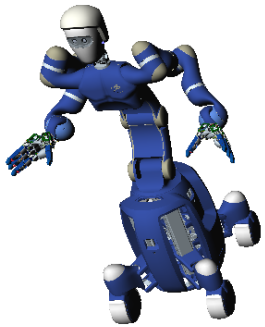
```
(:goal(and (on mug1 shelf) (on mug2 shelf) ))
```

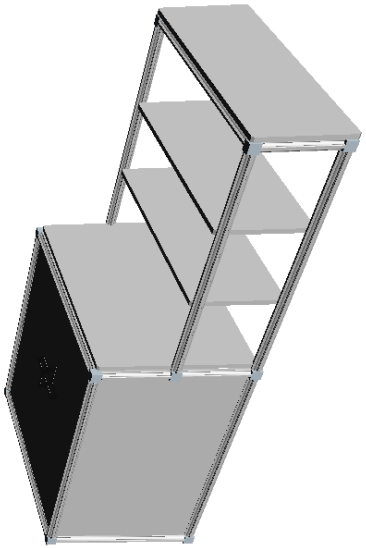
DLR

Base Positioning for Mobile Manipulation

Where should a mobile robot move to fulfill a task?

When does a mobile robot has to move its mobile base?



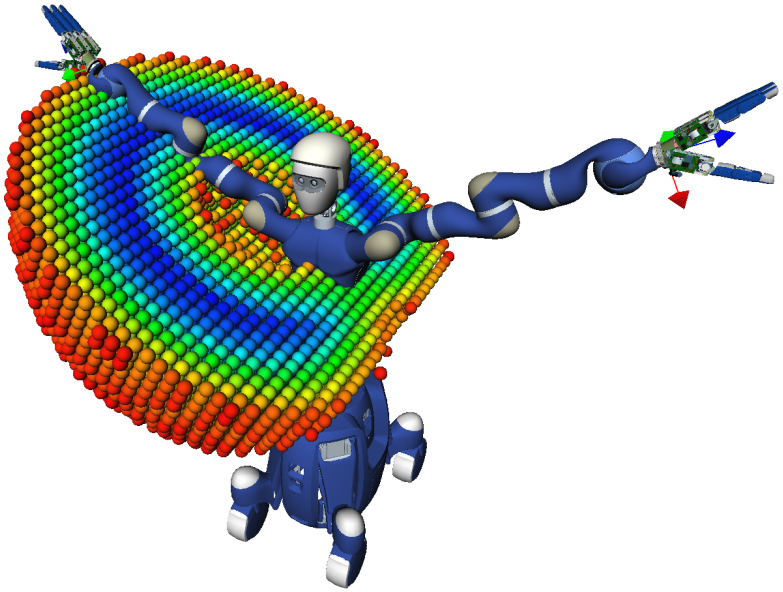


```
(:goal(and (on mug1 shelf) (on mug2 shelf) ))
```

DLR

Optimal Base Positioning

Capability Maps to represent reachability for robotic manipulators

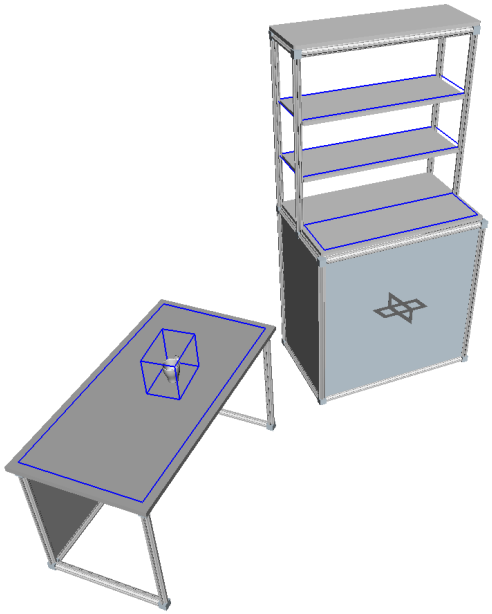


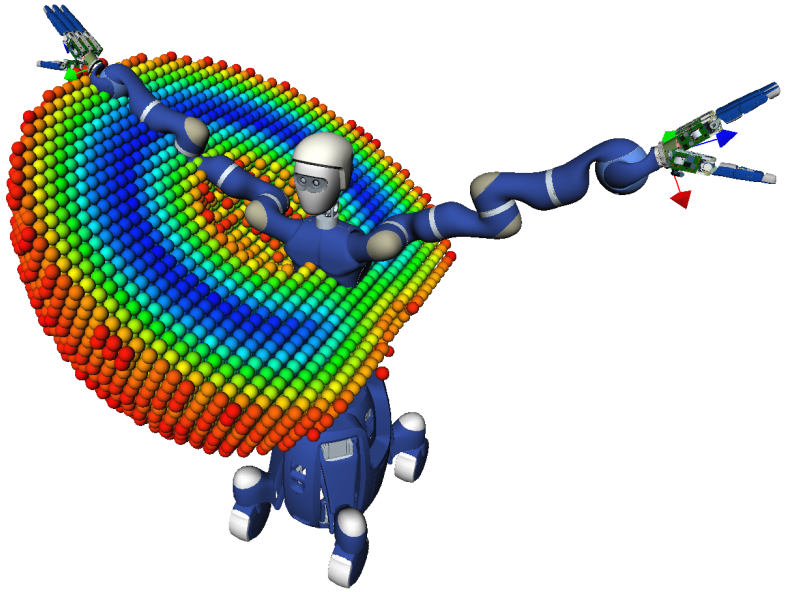
DLR

Optimal Base Positioning

Capability Maps to represent reachability for robotic manipulators, and *Regions of Interest (ROI)* based on object knowledge:

- Object specific: bounding box, table planes, storage positions, …
- Action specific: pick, place, open, stack, …



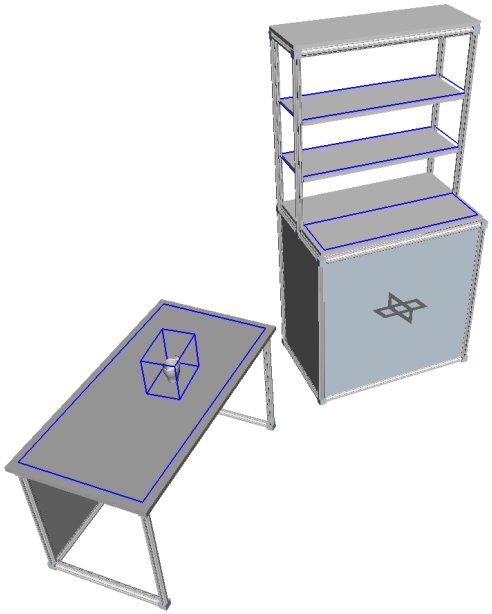


DLR

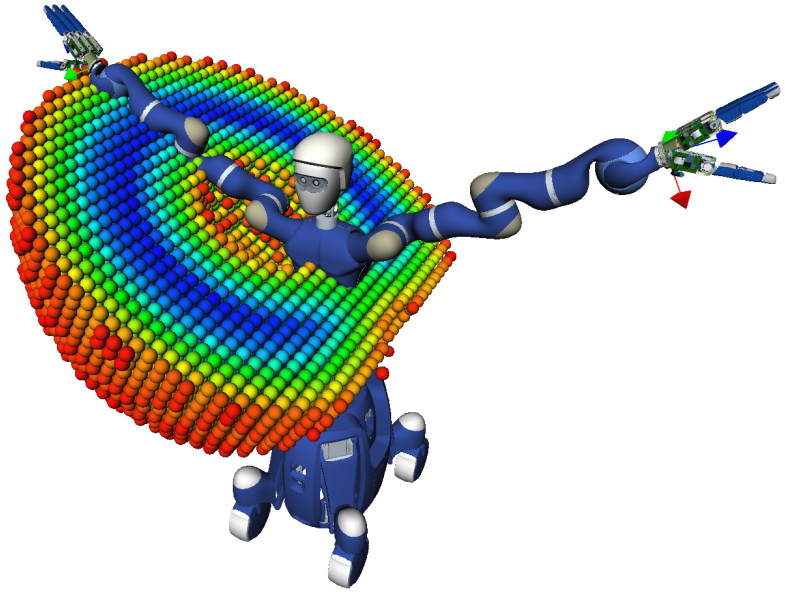
Optimal Base Positioning

Capability Maps to represent reachability for robotic manipulators, and *Regions of Interest (ROI)* based on object knowledge:

- Object specific: bounding box, table planes, storage positions, …
- Action specific: pick, place, open, stack, …



∩



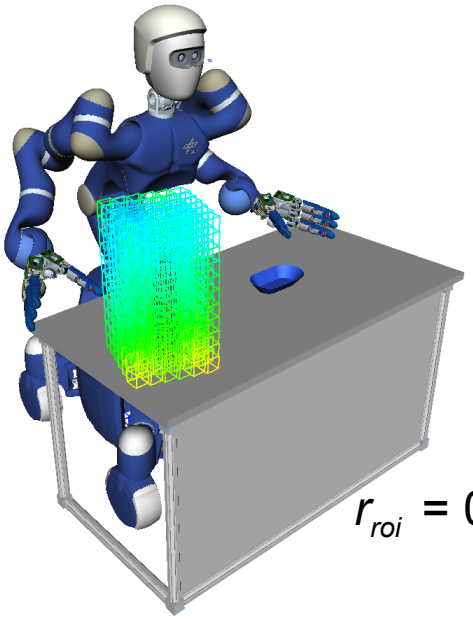
DLR

Optimal Base Positioning (cont.)

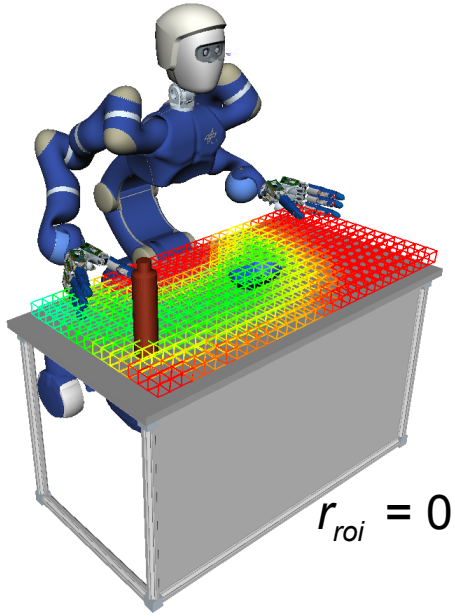
Optimizing the mean reachability r_{roi} within the given ROI:

$$\boldsymbol{x} = (\boldsymbol{p}_{\text{base}}^T, \boldsymbol{q}_{\text{torso}}^T)^T$$

$$\boldsymbol{x}_{\max} = \arg\max_{\boldsymbol{x}} \; r_{\text{roi}}(\boldsymbol{x})$$



r_{roi} = 0.66



r_{roi} = 0.32

Optimal Base Positioning (cont.)

A* Planner to navigate to the ROI center (unfeasible due to collision)

Algorithm: NAVIGATETOOBJECT($object$)

$map \leftarrow$ CREATEEMPTYMAP() # map initially empty
$reached \leftarrow False$
$roi \leftarrow object.roi$

while $reached \neq True$
do
- $path \leftarrow$ ASTARFINDPATH(map, roi) # get a-star path and check it for collision
- **if** CHECKCOLLISION($path$) $= False$
  - **then** $reached \leftarrow True$
  - **else**
    - UPDATEMAP(map) # update map on collision
    - **if** CHECKREACHABILITY(roi) ≥ 0.1
      - **then** $reached \leftarrow True$

OPTIMIZEREACHABILITY(roi) # optimize final position

Integrating Mobile Manipulation into Hybrid Reasoning

```
(:action place
 :parameters (?o - _object ?s - _tray ?m - _manipulator)
 :precondition (bound ?o ?m)
 :effect (and (on ?o ?s) (not(bound ?o ?m)) (free ?m))
)
```

DLR

Integrating Mobile Manipulation into Hybrid Reasoning

```
(:action place
 :parameters (?o - _object ?s - _tray ?m - _manipulator)
 :precondition (bound ?o ?m) (reachable ?o ?m)
 :effect (and (on ?o ?s) (not(bound ?o ?m)) (free ?m))
)
```

Geometric property for symbolic reasoning

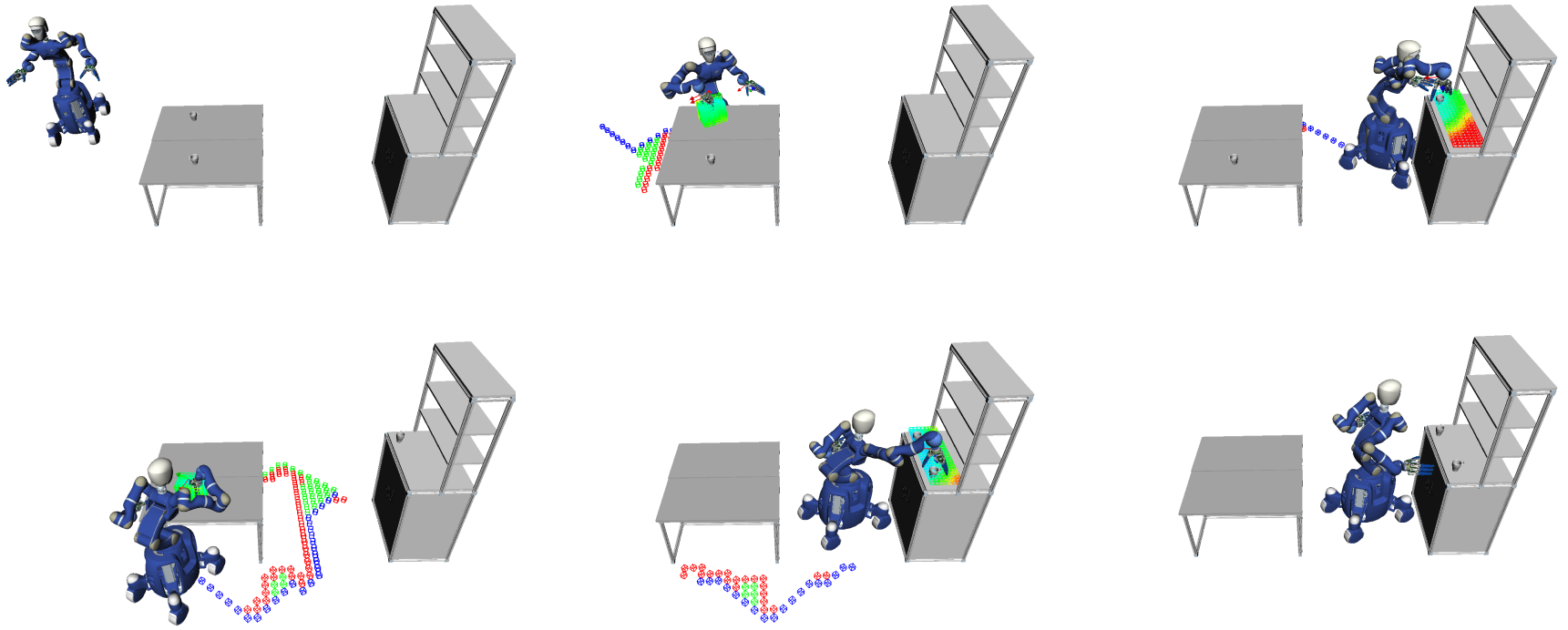
No geometric feedback in pure symbolic planning

→ Postpone decision to geometric reasoning step, if the current reachability for the ROI is beneath $r_{min} = r_{max} * 0.5$

DLR

Evaluation

Experiment in simulation where the humanoid robot Rollin' Justin has to place two mugs on a shelf



Discussion

Benefits and drawbacks of two step reasoning procedure

- inappropriate symbolic solution

+ less time consuming

+ modularity easier to handle

Benefits and drawbacks of separating navigation and motion planning

- generality decreases

+ suitable for larger goal regions

DLR

Conclusion

Hybrid reasoning based on object knowledge

- Action templates to describe process models
- Integrated geometric backtracking

Optimal base positioning for mobile manipulation

- Intersection between capability maps and object ROI

Integrating mobile manipulation into hybrid reasoning

- Geometric reasoning without symbolic overhead

Evaluation in a simulated experiment

- Experiments on the real robot in the near future

DLR

Thank you for your attention!