A Deliberation Layer for Instantiating Robot Execution Plans from Abstract Task Descriptions

June 10th, 2013

D. Di Marco[*], R.J.M. Janssen[°], A.C. Perzylo[⊛], M.J.G. van de Molengraft[°], P. Levi[*]

[°] Eindhoven University of Technology, The Netherlands
[⊛] Technische Universität München, Germany
[*] Universität Stuttgart, Germany

dimarco@ipvs.uni-stuttgart.de

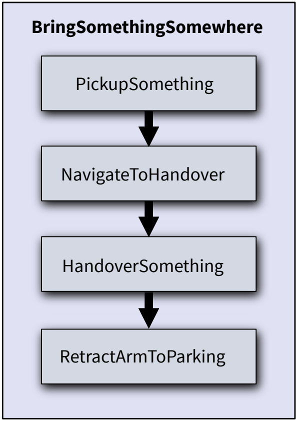
Motivation

Primary question: How can task descriptions be shared among different service robot platforms in different environments?

- Represent task descriptions in an abstract way (**action recipe**)
- Decouple from environment & robot hardware specifics
- Store information in a globally accessible database

How use these to generate execution plans tailored to the current situation?

Action Recipe

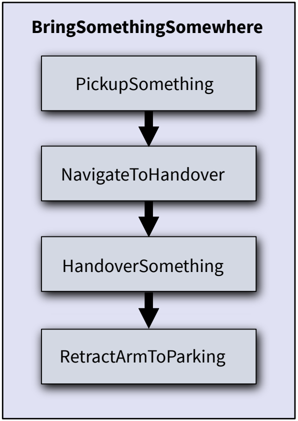


Abstract (i.e. independent of robot platform and environment) task descriptions[1]

- Formulated in **OWL**, based on a common ontology
- Can be composed of other recipes or primitive actions, sub-actions are partially ordered
- Annotated with description of requirements, to check whether a given robot is capable of executing these

[1] M. Tenorth et al.: "The RoboEarth language: Representing and Exchanging Knowledge about Actions, Objects, and Environments", ICRA 2012

Action Recipe



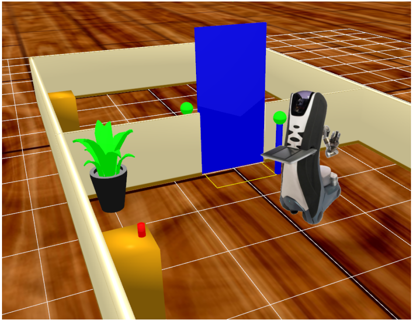
Abstract (i.e. independent of robot platform and environment) task descriptions[1]

Example Recipe

```
Class: BringSomethingSomewhere
  EquivalentTo:
    (knowrob:toLocation some
        robotPose-handover1)
  SubClassOf:
    roboearth:IntentionalAction
...
Individual: ServeADrinkOrder10
  Types:
    knowrob:PartialOrdering-Strict
  Facts:
    knowrob:occursBeforeInOrdering MoveBaseToGraspPose,
    knowrob:occursAfterInOrdering GraspBottle
```

[1] M. Tenorth et al.: ”The RoboEarth language: Representing and Exchanging Knowledge about Actions, Objects, and Environments”, ICRA 2012

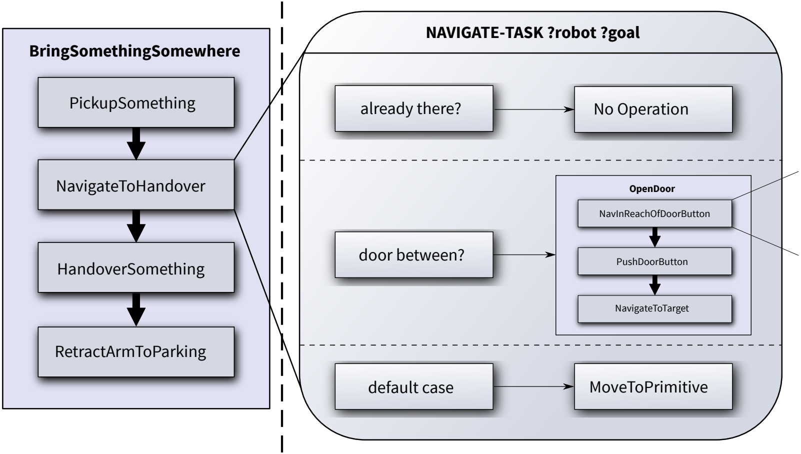
Action Instantiation



- Task: navigate into the next room
- How to infer that ”Navigate” has to be instantiated as ”open door, pass open door”?

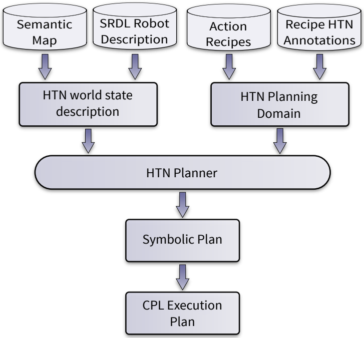
⇒ we need more knowledge

Action Recipe Annotation



HTN decompositions for actions with preconditions & effects, fully encoded in OWL and linked to recipes.

Plan Instantiation Process

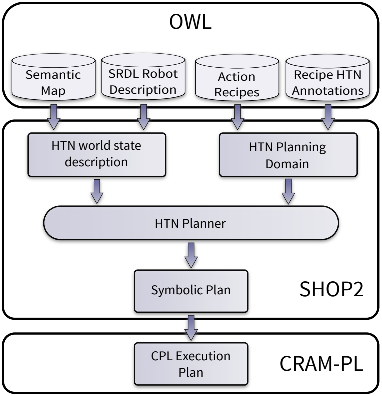


- Semantic Maps: basic topology of environment
- SRDL: Semantic Robot Description Language[1]
- Off-the-shelf HTN planner: SHOP2[1]
- CRAM Plan Language (CPL) to specify execution plans

[1] L. Kunze et al.: "Towards semantic robot description languages". ICRA 2011

[2] D. Nau et al.: "SHOP2: An HTN planning system". JAIR, 2003

Plan Instantiation Process



- Semantic Maps: basic topology of environment
- SRDL: Semantic Robot Description Language[1]
- Off-the-shelf HTN planner: SHOP2[1]
- CRAM Plan Language (CPL) to specify execution plans

[1] L. Kunze et al.: "Towards semantic robot description languages". ICRA 2011
[2] D. Nau et al.: "SHOP2: An HTN planning system". JAIR, 2003

CRAM Plan Language[1]

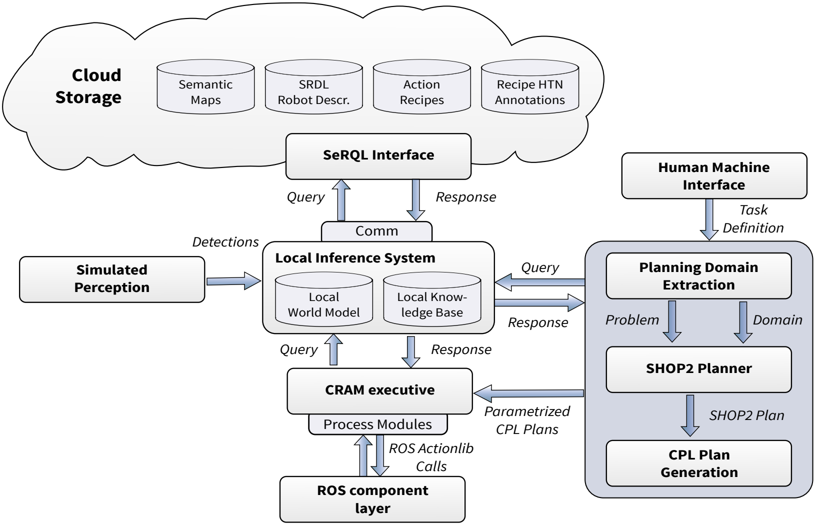
Extension of Common Lisp for robot execution plans

- Control structures for executing/synchronizing tasks in parallel
- Implementation of the Rete algorithm for Prolog-like reasoning facilities
- **Designators**: symbolic description of objects, locations, task parts
    - Based on ideas from Firby[2]
    - Created from action specifications in recipe
    - Get resolved at the latest possible instant during plan execution
    - Independent of **Hardware/Environment**

[1] M. Beetz, L. Mösenlechner, M. Tenorth, "CRAM - A Cognitive Robot Abstract Machine for Everyday Manipulation in Human Environments". IROS, 2010

[2] R.J. Firby, "Adaptive execution in complex dynamic worlds". PhD thesis, Yale University, 1989

Architecture



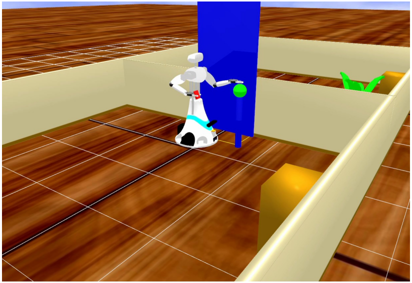
Simulation Experiment

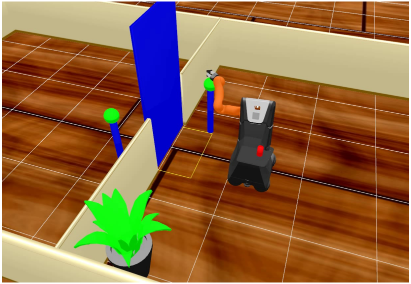
- Gazebo simulator[1] used
- Fraunhofer IPA's Care-o-Bot 3-4 (1 arm, 1 tray) and TU/e's Amigo (2 arms) prototype
- Task description: "bring something somewhere"

- Both robots use same top-level recipe, but need to select different decompositions

[1] http://gazebosim.org

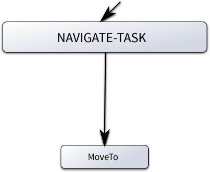
Experiment Environment

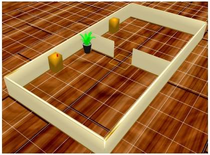




The door can be opened by touching the green button. The used arm has to be free.

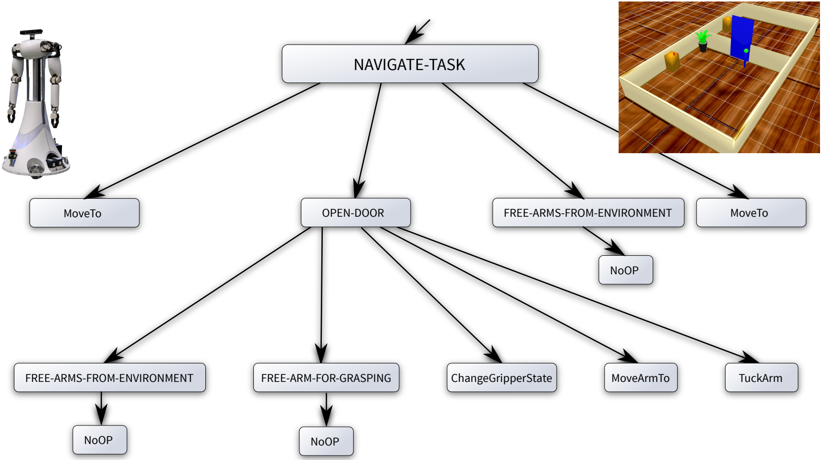
Differences in generated Plans





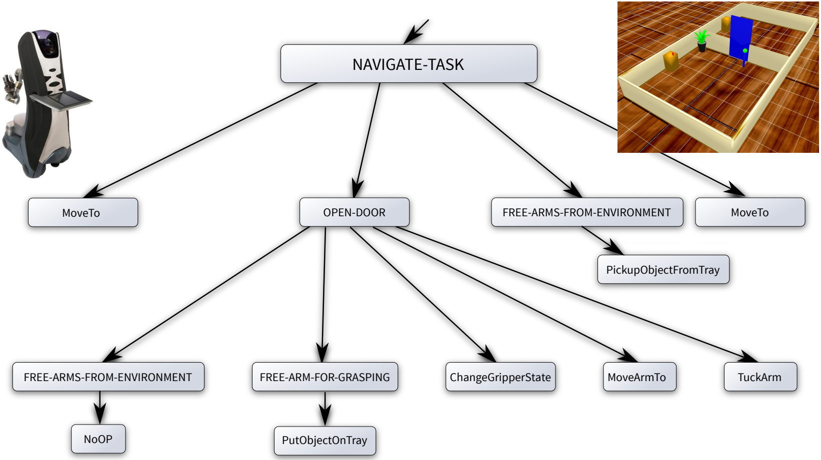
No door: navigation between rooms trivial

Differences in generated Plans



2 manipulators: grasp object with left, open door with right arm

Differences in generated Plans



1 manipulator + tray: put object on tray, open door, pickup object

Video

- Video Demo -

Conclusion

- Increase applicability of abstract task descriptions using off-the-shelf HTN planning
- Separation of sources of knowledge
- Experiment required about 4K lines of OWL
- Classic symbolic reasoning, e.g. no temporal aspects

Future work:

- Apply approach to more difficult scenarios
- Integrate plan generation and task execution for more robust behavior

Discussion

Thank you for your attention