

Learning not to fail

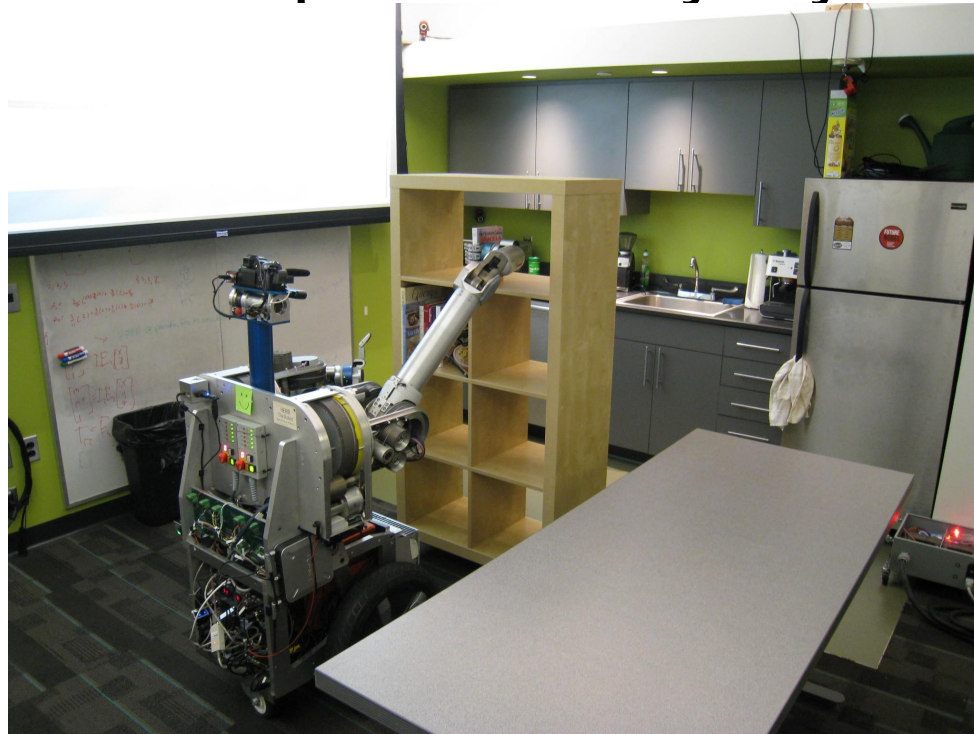
Improving task execution with experience

Andrey Kurenkov

Personal Robotics Lab
19 July 2013

Motivation

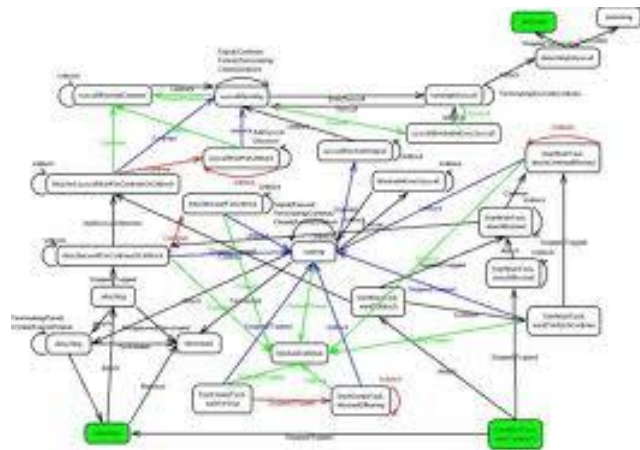
Robots able to help with everyday tasks



HERB

Problems

Describing tasks

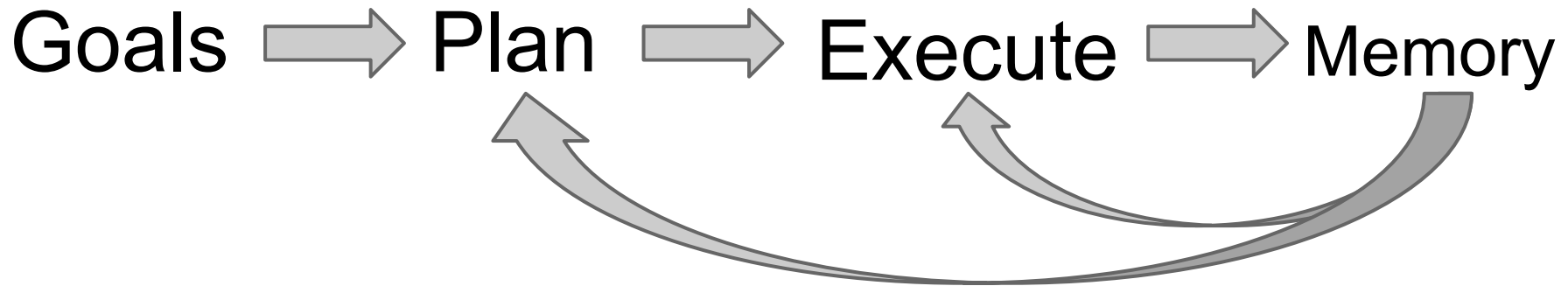


Handling error



Hypothesis

**Need to reason about actions given goals
and learn from past experience**



Related work



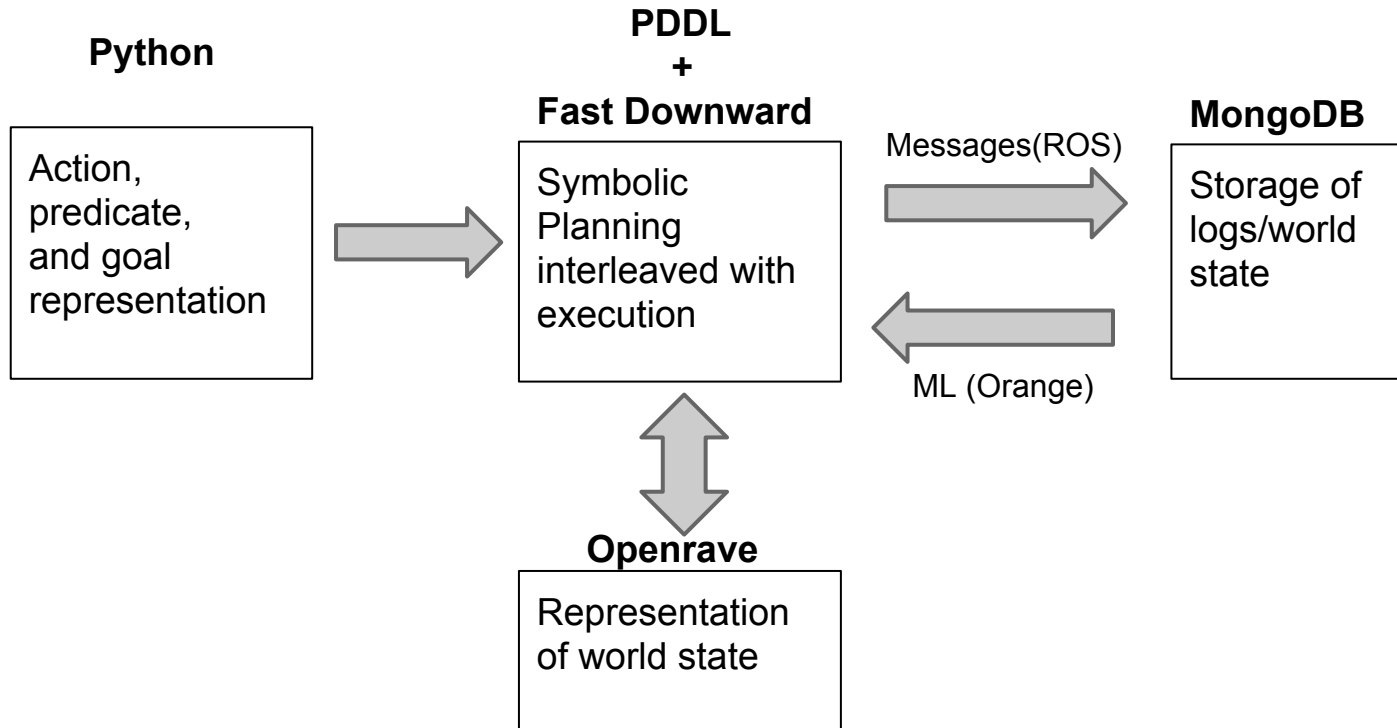
Shakey

STRIPS and successors

Continuous + symbolic reasoning

Learning to modify planning

Technical Approach



PDDL

Standard description language for AI task planning

PDDL demo programming:

action and predicate library +

one or more goals =

herb does stuff (with some notion of what is happening)

PDDL Pseudocode

```
while not state.is_goal():  
    plan = generate_plan(problem,state)  
    action = plan[0]  
    if not preconditions_true(action,state):  
        remove_false_preconditions(action,state)  
        continue  
    state = state + action.execute()  
    if not postconditions_true(action,state):  
        log failure  
else  
    log success
```

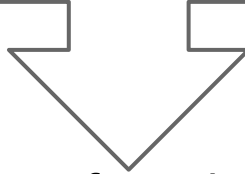
PDDL Librarian demo

Initial state:

aware_of_object(bookcase) & obj_at(Herb2,bookcase_place) & obj_in(dracula,bookcase) & is_loc(test_loc3) & is_loc(bookcase_place) & aware_of_object(dracula) & is_clear(bookcase) & aware_of_object(Herb2) & herb_loc_known() & is_facing(dracula) & is_small(dracula) & is_facing(Herb2) & aware_of_object(table) & is_clear(dracula) & herb_clear() & is_clear(Herb2) & is_loc(test_loc2) & is_loc(handoff) & is_clear(table) & herb_at(bookcase_place)

Goals:

**obj_handed_at('dracula','handoff'),
obj_on('dracula','desk'),
obj_in('dracula','bookcase')**



Plan: grasp_from_bookshelf('dracula'),
goto_place_obj_at('dracula','handoff')
hand_obj_at('dracula','handoff')
teleport_book('dracula','table')
edge_grasp('dracula','table')
goto('bookcase_place')
place_in_bookcase('dracula')

PDDL Librarian demo



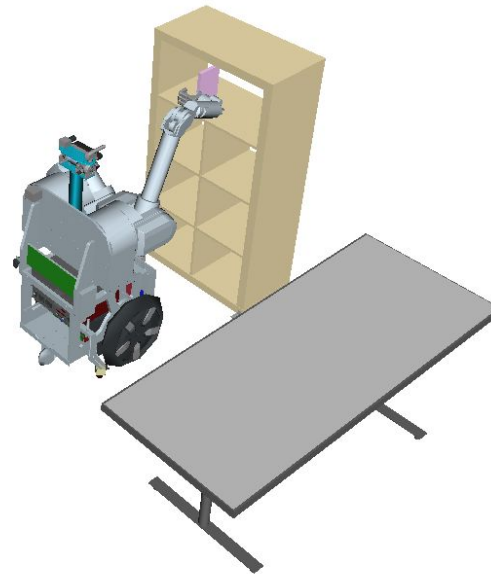
```
{
  'urun_id': 1374184517558L, 'ugoal': 'u'herb_has(dracula)', 'ulevel': 'uINFO', 'urobot_tag': 1374184517584L, 'u'
    'r': 1374184517558, 'u'bookcase': 1374184517558, 'udracula': 1374184517558}', 'u'_topic': 'u/pddl/logs', 'u'line
    'racula' & is_clear(Herb2) & is_loc(test_loc2) & is_loc(handoff) & obj_in(dracula,bookcase) & aware_of_object(
      oc) & herb_at(bookcase_place) & aware_of_object(dracula) & aware_of_object(Herb2) & is_loc(bookcase_place) &
      c3')', 'u'_id': ObjectId('51e8650b0cf3f62e9ad5258'), 'u''_recorded': datetime.datetime(2013, 7, 18, 17, 58, 35),
    {'urun_id': 1374184517558L, 'ugoal': 'u'herb_has(dracula)', 'ulevel': 'uINFO', 'urobot_tag': 1374184517585L, 'u'
      e': 1374184517558, 'u'bookcase': 1374184517558, 'udracula': 1374184517558}', 'u'_topic': 'u/pddl/logs', 'u'line
      dracula' & is_clear(Herb2) & is_loc(test_loc2) & is_loc(handoff) & obj_in(dracula,bookcase) & aware_of_object(
        loc) & herb_at(bookcase_place) & aware_of_object(dracula) & aware_of_object(Herb2) & is_loc(bookcase_place) &
        c3')', 'u'_id': ObjectId('51e8650c0cf3f62e9ad5259'), 'u''_recorded': datetime.datetime(2013, 7, 18, 17, 58, 36),
    {'urun_id': 1374184517558L, 'ugoal': 'u'erhd_has(dracula)', 'ulevel': 'uINFO', 'urobot_tag': 1374184517585L, 'u'
      'table': 1374184517558, 'u'bookcase': 1374184517558, 'udracula': 1374184517558}', 'u'_topic': 'u/pddl/logs', 'u'line
      lear(dracula) & is_clear(Herb2) & is_loc(test_loc2) & is_loc(handoff) & obj_in(dracula,bookcase) & aware_of_o
        test_loc) & herb_at(bookcase_place) & aware_of_object(dracula) & aware_of_object(Herb2) & is_loc(bookcase_pla
          est_loc3')', 'u'_id': ObjectId('51e8650c0cf3f62e9ad525a'), 'u''_recorded': datetime.datetime(2013, 7, 18, 17, 58
            }
    {'urun_id': 1374184517558L, 'ugoal': 'u'herb_has(dracula)', 'ulevel': 'uINFO', 'urobot_tag': 1374184517585L, 'u'
      u'table': 1374184517558, 'u'bookcase': 1374184517558, 'udracula': 1374184517558}', 'u'_topic': 'u/pddl/logs',
        clear(dracula) & is_clear(Herb2) & is_loc(test_loc2) & is_loc(handoff) & obj_in(dracula,bookcase) & aware_of
          (test_loc) & herb_at(bookcase_place) & aware_of_object(dracula) & aware_of_object(Herb2) & is_loc(bookcase_pl
            est_loc3')', 'u'_id': ObjectId('51e8650d0cf3f62e9ad525b'), 'u''_recorded': datetime.datetime(2013, 7, 18, 17, 5
              e}
    {'urun_id': 1374184517558L, 'ugoal': 'u'herb_has(dracula)', 'ulevel': 'uINFO', 'urobot_tag': 1374184517586L, 'u'
      'goal': array([ 3.12961015, -1.098177 , 0.1 , 1.38481888, -1.57844219,\n -1.46949108, -0.3226
        84517558}], 'u'_topic': 'u/pddl/logs', 'u'line_num': 366, 'u'predicates': 'u'herb_loc_known()' & is_facing(bookca
          ) & obj_in(dracula,bookcase) & aware_of_object(bookcase) & is_facing(dracula) & obj_at(Herb2,bookcase_place)
            re_of_object(Herb2) & is_loc(bookcase_place) & herb_clear() & is_clear(bookcase) & is_small(dracula) & aware_
              d': datetime.datetime(2013, 7, 18, 17, 58, 37, 484000), 'u'action': 'u'grasp_from_bookshelf(objname)[objname]
                {'urun_id': 1374184517558L, 'ugoal': 'u'herb_has(dracula)', 'ulevel': 'uINFO', 'urobot_tag': 1374184517586L, 'u'
                  4517558, 'u'bookcase': 1374184517558, 'udracula': 1374184517558}', 'u'_topic': 'u/pddl/logs', 'u'line_num': 44,
                    is_clear(Herb2) & is_loc(test_loc2) & is_loc(handoff) & obj_in(dracula,bookcase) & aware_of_object(bookcase)
                      b_at(bookcase_place) & aware_of_object(dracula) & aware_of_object(Herb2) & is_loc(bookcase_place) & herb_clea
                        d': ObjectId('51e8651400cf3f62e9ad525d'), 'u''_recorded': datetime.datetime(2013, 7, 18, 17, 58, 44, 134000),
                          {'urun_id': 1374184517558L, 'ugoal': 'u'herb_has(dracula)', 'ulevel': 'uINFO', 'urobot_tag': 1374184517586L, 'u'
                            ll', 'u'env_tags': 'u'(u'table': 1374184517558, 'u'bookcase': 1374184517558, 'udracula': 1374184517558)}, 'u'_to
                              is_clear(table) & is_clear(dracula) & is_clear(Herb2) & is_loc(test_loc2) & is_loc(handoff) & obj_in(dracula
                                facing(Herb2) & is_loc(test_loc) & herb_at(bookcase_place) & aware_of_object(dracula) & aware_of_object(Herb2)
                                  ject(table) & is_loc(test_loc3')', 'u'_id': ObjectId('51e8651400cf3f62e9ad525e'), 'u''_recorded': datetime.datetime(2013, 7, 18, 17, 58, 44, 134000),
                                    acula'])), 'u'sim': False}
    {'urun_id': 1374184517558L, 'ugoal': 'u'herb_has(dracula)', 'ulevel': 'uINFO', 'urobot_tag': 1374184517587L, 'u'
      'env_tags': 'u'(u'table': 1374184517558, 'u'bookcase': 1374184517558, 'udracula': 1374184517558)}, 'u'_topic':
        ar(table) & is_clear(dracula) & is_clear(Herb2) & is_loc(test_loc2) & is_loc(handoff) & obj_in(dracula,bookca
          Herb2) & is_loc(test_loc) & herb_at(bookcase_place) & aware_of_object(dracula) & aware_of_object(Herb2) & is_
            ble' & is_loc(test_loc3')', 'u'_id': ObjectId('51e8651900cf3f62e9ad525f'), 'u''_recorded': datetime.datetime(201
              , 'u'sim': False}
    {'urun_id': 1374184517558L, 'ugoal': 'u'herb_has(dracula)', 'ulevel': 'uINFO', 'urobot_tag': 1374184517587L, 'u'
      'r': 1374184517558, 'u'bookcase': 1374184517558, 'udracula': 1374184517558}', 'u'_topic': 'u/pddl/logs', 'u'line
        racula' & is_clear(Herb2) & is_loc(test_loc2) & is_loc(handoff) & obj_in(dracula,bookcase) & aware_of_object(
          oc) & herb_at(bookcase_place) & aware_of_object(dracula) & aware_of_object(Herb2) & is_loc(bookcase_place) &
          c3')', 'u'_id': ObjectId('51e8651900cf3f62e9ad5260'), 'u''_recorded': datetime.datetime(2013, 7, 18, 17, 58, 49),
    {'urun_id': 1374184517558L, 'ugoal': 'u'erhd_has(dracula)', 'ulevel': 'uINFO', 'urobot_tag': 1374184517588L, 'u'
      e': 1374184517558, 'u'bookcase': 1374184517558, 'udracula': 1374184517558}', 'u'_topic': 'u/pddl/logs', 'u'line
        dracula' & is_clear(Herb2) & is_loc(test_loc2) & is_loc(handoff) & obj_in(dracula,bookcase) & aware_of_object(
          loc) & herb_at(bookcase_place) & aware_of_object(dracula) & aware_of_object(Herb2) & is_loc(bookcase_place) &
          c3')', 'u'_id': ObjectId('51e8651900cf3f62e9ad5261'), 'u''_recorded': datetime.datetime(2013, 7, 18, 17, 58, 49,
```

Database uses

Simulated replay of executions

Evaluation of new predicates over past executions

Modifying task planning



Detection and handling of reasons for action failure

- ie grasp_from_bookshelf fails if robot angle > 1.2 at start of action

Considerations

1. Need to reveal cause of error
2. Many ways to build feature vector
3. Learning within a range of continuous time
4. Incorporating learning into execution

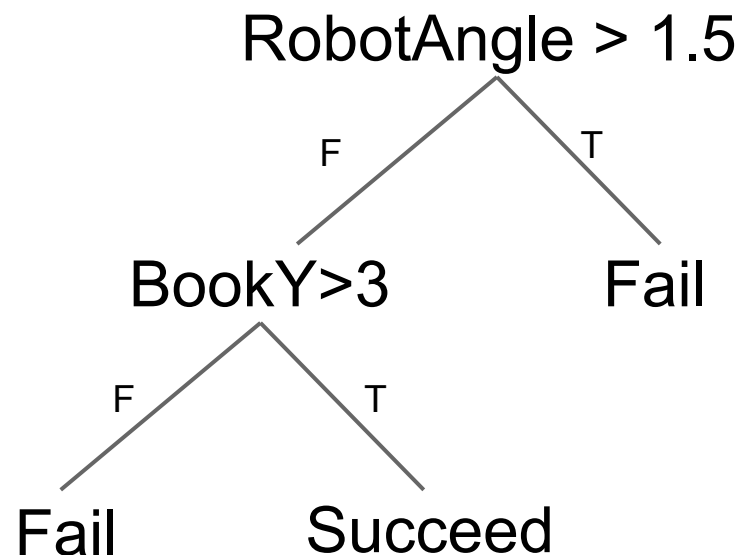
Causes of error

Assumption: many errors can be avoided by learning constraints on single physical variables

Classification tree

simple constraints

built-in feature selection



Feature Vector

1D physical variables (x, y, z, θ) of pertinent world objects: robot, end effectors, objects

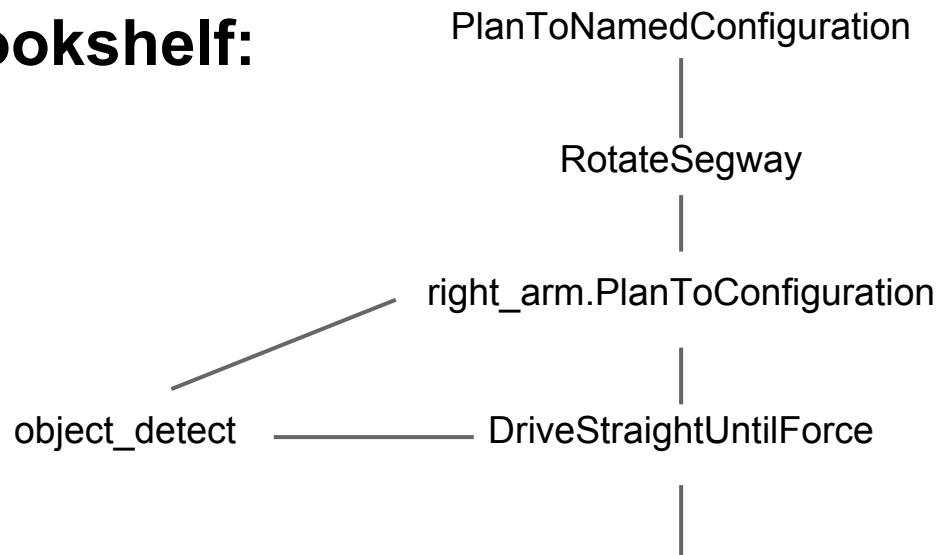


Temporal learning

Train error classifier for every 'sub-action'
within each PDDL action

- execution order consistent

grasp_from_bookshelf:



Future Work

Filling out the database

Creating of valid simulation action to learn on

Considering other ML approaches

Incorporating learning back into execution