Plan-based Action and Activity Control for Everyday Manipulation

Michael Beetz and the IAS Group
Intelligent Autonomous Systems
Technische Universität München

ICRA Workshop
Motion Planning for Physical Robots
Shanghai, May 2011
Our Vision:
Cognitive robots that
- autonomously perform human-scale everyday manipulation tasks and
- extend their repertoire of such by acquiring new skills using information resources intended for human use.
Robotic roommates making “Weisswürste”

Understanding by building

Shopping & cleaning up

1. shopping with basket

   ![Image of robot shopping with basket]

2. clean up according to organizational principles

   ![Image of robot cleaning up]

Making “Weisswürste”

1. putting “Weisswürste” into pot

   ![Image of robot putting “Weisswürste” into pot]

2. fishing “Weisswürste”

   ![Image of robot fishing “Weisswürste”]

3. cutting bread

   ![Image of robot cutting bread]
Research Questions

▶ how is it possible that we as humans get instructions such as
  ▶ make pancake using a pancake mix
  ▶ flip the pancake
  ▶ push the spatula under the pancake
  and perform the intended tasks in the appropriate way?

▶ What are computational models that enable robots to perform such naturalistic action specifications?
Everyday Manipulation Activities

From:
- mix flour and milk
- crack the egg
- mix the egg yolk with the dough
- pour the dough onto the pancake maker
- flip the pancake

To:
- take egg and milk from fridge
- take flour from cupboard
- mix flour and milk
- crack the egg
- mix the egg yolk with the dough
- pour the dough onto the pancake maker
- baking
- flip the pancake
- baking

Robot Plan Generation

Motion Planning for Physical Robots

Everyday Manipulation

Michael Beetz
Inferring required objects

- **Predict intermediate objects**: Projection of action effects
- **Make sure they can be recognized**: Download object models

Motion Planning for Physical Robots

Everyday Manipulation
Inferring where objects can be found

▶ Infer where to look for objects: Semantic environment models
▶ Add actions to fetch them: Articulation models and observations of humans

Motion Planning for Physical Robots

Everyday Manipulation
Planning with processes

- Check if results are correct: Action effect axioms
- Add required actions: Planning with actions and processes

Motion Planning for Physical Robots

Everyday Manipulation

Michael Beetz
Choose action parameters: Physically simulate action effects
From: “push the spatula under the pancake"

Task Specification

Define cylinder+rpy coordinates $\langle \theta, x, z \rangle, \langle r, p, y \rangle$ between spatula and pancake.

- move in horizontal direction towards pancake ($x = 0$)
- keep spatula horizontal ($r = 0, p < 10^\circ$)
- maintain contact force with pancake baker ($\text{force}(z) > 1N$)
3 Layer Architectures

Abstract (symbolic) layer

“push the spatula under the pancake”

Low-level control

collectors, dynamic motion primitives, motion plans, ...
3 Layer Architectures

Abstract (symbolic) layer
“push the spatula under the pancake"

Plan execution layer
bridges between high- and lowlevel control

Low-level control
controllers, dynamic motion primitives, motion plans, ...
Three fundamental research questions:

- How to specify actions?
- How to get what is meant from what is specified?
- How to execute what is meant?
(perform (an action
  (attribute/constraint\_1 value\_1)
  ...
  (attribute/constraint\_n value\_n)

<table>
<thead>
<tr>
<th><strong>vague specification</strong></th>
<th><strong>effective specification</strong></th>
</tr>
</thead>
</table>
| push the spatula under the pancake | push the spatula under the pancake such that
  ○ you can lift the pancake safely,
  ○ don’t damage the pancake, and
  ○ don’t push the pancake off the oven |
### Naturalistic Action Specifications

**What is meant**

<table>
<thead>
<tr>
<th>what is specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>push the spatula under the pancake</td>
</tr>
<tr>
<td>flip the pancake carefully</td>
</tr>
<tr>
<td>put the pancake mix down</td>
</tr>
<tr>
<td>take the spatula</td>
</tr>
</tbody>
</table>
### Naturalistic Action Specifications

#### What is meant

<table>
<thead>
<tr>
<th>what is specified</th>
<th>what is meant</th>
</tr>
</thead>
</table>
| push the spatula under the pancake             | push the spatula under the pancake such that  
  ○ you can lift the pancake safely,  
  ○ don’t damage the pancake, and  
  ○ don’t push the pancake off the oven          |
| flip the pancake carefully                     | flip the pancake such that  
  ○ undesired side effects are avoided and  
  ○ the robot stops if they might happen          |
| put the pancake mix down                       | put the pancake mix down where  
  ○ it is visible and reachable when needed and  
  ○ it does not hinder the overall activity       |
| take the spatula                               | take the  
  ○ handle of the spatula  
  ○ such that you can perform precision control of the blade of the spatula                                                             |
Naturalistic Actions

Building an action library

Mining instructions from wikiHow.com

- 273 Categories
- 8786 NL-Plans
- >130,000 sentences
- ≈53,000 relevant instructions
- ≈100 relevant action verbs
- most important: adding sth (> 7,900), Picking/Placing (> 4,900)
- top 15 action verbs more than 50% of actions

Food-and-Entertaining

Recipes

- Pasta-Sauce-Recipes
- Spaghetti
- Lasagna
- Macaroni-and-Cheese
- Drying-Food
- Freezing-Food
- Fast-Food
- Spaghetti
- Salads
- Pasta-and-Noodles
- Soups
- Pork
- Beef-and-Lamb
- Poultry
- Drinks
- Barbecue
- Baking
- Fruits-and-Vegetables
- Eggs-and-Dairy
- Desserts-and-Sweets
- Fish-and-Seafood-Sandwiches
- Fish-and-Seafood
- Sandwiches-and-Quick-Meals
- Peanut-Butter-Sandwiches
- Meat
- Meat-and-Poultry-Sandwiches
- Basic-Cooking-Skills
- Food-Preservation-Techniques
- Food-Cutting-Techniques
- Drying-Food
- Freezing-Food
- Toast
- Cheese-Sandwiches
- Breakfast-Cereal
- Pancakes
- Waffles
- Granola-and-Muesli
- Oatmeal-and-Porridge
- Breakfast
- Fried-Eggs
- Scrambled-Eggs
- Omelette

Motion Planning for Physical Robots

Everyday Manipulation

Michael Beetz
Flipping

Building an action ontology
The semantic core of action verbs

Semantic Core: Set of inter- and intraconceptual relations that constitute an abstract event type, assigning a semantic role to each entity that is affected by an action verb.

(thanks to Torsten Schubert)
The Semantic Core of Action Verbs

Action Specification

**Push** _ActionVerb_ the spatula _Instrument_ under _Place_ the pancake _Ground_ _LocativeRelation_ Agent

---

Motion Planning for Physical Robots

Everyday Manipulation

Michael Beetz
Take the spatula
Automatic completion of action verbs

(perform (an action
    (type grasp)
    (object (an object-part
        (part-of spatula)
        (type handle)))
    (desired-effect (and (grasp ?grasp-spec)
        (succeeds (an action
            (type precision-control)
            (object (tip-of blade))
            (grasp ?grasp-spec)))))))

probabilistic completion

P(occurs(ev),
    type(ev,type),
    bodyPartOfAgent(ev,Part),
    entity(ev,what)
| “push the spatula under the pancake”)

tuple with the highest probability:
- type = holding
- bodyPartOfAgent = hand
- entity = handle
Push the spatula under the pancake

Consequence-based action parameterization

(perform (an action
  (type push)
  (object (an object-part
    (part-of spatula)
    (type blade)))
  (destination ?loc = (a location
    (under pancake)))
  (desired-effect (and (pose spatula ?loc)
    (succeeds (an action
      (type lift)
      (object pancake)
      (starting-pose ?loc))))
    (undesired-effects (and (damaged pancake)
      (on pancake counter))))

Motion Planning for Physical Robots
Temporal Projection Process

- **make_projection**: sample parameters
  - **simulate**: setup simulator
  - run simulation
  - **translate**: ground predicates in logged simulations
  - **evaluate**: events/fluents specialized predicates

```
make_projection(scenario, program, params, Timeline)

occur(entry, Time, Timeline)

egg slips away
egg is held in gripper
egg is broken
```

Motion Planning for Physical Robots

Everyday Manipulation

Michael Beetz
Example: Pouring pancake mix

- **Parameters:** position, time, angle
- **Outcomes:** number of particles on pan (spilled on table)

- Specialized predicates on particle sets: **round/centered**
Example: Flipping a pancake

- **Parameters:** angle of spatula
- **Outcomes:** turned, not turned

- **Common failures:**
  - break
  - push off
  - fold
  - stick on
Put the pancake mix away

Exploiting underspecification

(perform (an action
  (type put-away)
  (object ?obj = (the object
    (type pancake-mix) ))
  (destination ?loc = (a location
    (on counter)
    (stable ?obj)
    (reachable t)
    (visible-for James)
    (not (hindering (the activity
      (type pancake-making)))))))))
Inference algorithm

\[
\text{setof } \text{?Pose On(Counter, ?Pose) } \text{?Poses } \land \text{ member(} \text{?P, ?Poses) } \\
\land \text{ Pose(Cup, ?P) } \land \text{ stable(Cup)}
\]
Inference algorithm

\[
\text{setof } \text{?Pose } \text{On(Counter, } \text{?Pose) } \text{?Poses } \land \text{ member(} \text{?P, } \text{?Poses) } \\
\land \text{ Pose(Cup, } \text{?P) } \land \text{ stable(Cup)}
\]

1. \text{setof } \text{?Pose } \text{On(Counter, } \text{?Pose) } \text{?Poses}
2. \text{member(} \text{?P, } \text{?Poses)}
3. \text{Pose(Cup, } \text{?P)}
4. \text{stable(Cup)}

Create distribution for sampling poses
Inference algorithm

\[ \text{setof } \text{?Pose } \text{On(Counter, ?Pose) } \text{?Poses } \land \text{member(?P, ?Poses)} \land \text{Pose(Cup, ?P)} \land \text{stable(Cup)} \]

1. setof ?Pose On(Counter, ?Pose) ?Poses
2. member(?P, ?Poses)
3. Pose(Cup, ?P)
4. stable(Cup)

Draw a pose sample
Inference algorithm

\[
\text{setof } \text{?Pose On(Counter, ?Pose) } \text{?Poses } \land \text{member(?P, ?Poses) } \\
\land \text{Pose(Cup, ?P) } \land \text{stable(Cup)}
\]

1. setof ?Pose On(Counter, ?Pose) ?Poses
2. member(?P, ?Poses)
3. Pose(Cup, ?P)
4. stable(Cup)

Place the mug
Inference algorithm

\[
\text{setof } \text{?Pose } \text{On(Counter, ?Pose)} \text{ ?Poses} \land \text{member(?P, ?Poses)} \\
\land \text{Pose(Cup, ?P)} \land \text{stable(Cup)}
\]

1. setof ?Pose On(Counter, ?Pose) ?Poses
2. member(?P, ?Poses)
3. Pose(Cup, ?P)
4. stable(Cup)

Simulate for 50ms, fail!

1. setof ?Pose On(Counter, ?Pose) ?Poses
2. member(?P, ?Poses)
3. Pose(Cup, ?P)
4. stable(Cup)

Backtrack, draw another pose sample

1. setof ?Pose On(Counter, ?Pose) ?Poses
2. member(?P, ?Poses)
3. Pose(Cup, ?P)
4. stable(Cup)
Inference algorithm


1. setof ?Pose On(Counter, ?Pose) ?Poses
2. member(?P, ?Poses)
3. Pose(Cup, ?P)
4. stable(Cup)

Simulate for 50ms, succeed!
### Built-in Predicates of CRAM Reasoning

<table>
<thead>
<tr>
<th>Stability</th>
<th>Visibility</th>
<th>Reachability</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{contact}(O_1, O_2)</td>
<td>\textit{visible}(P, O)</td>
<td>\textit{reachable}(R, O)</td>
</tr>
<tr>
<td>stable(O)</td>
<td>\textit{occluding}(P, O_1, O_2)</td>
<td>\textit{blockingObjects}(R, O, B)</td>
</tr>
<tr>
<td>Contact between objects</td>
<td>Object visible from pose P</td>
<td>Object O is reachable by robot R</td>
</tr>
<tr>
<td>Stability of object</td>
<td>Object (O_2) occludes object (O_1)</td>
<td>B is the list of objects that “block” O</td>
</tr>
</tbody>
</table>

**Motion Planning for Physical Robots**

Everyday Manipulation

Michael Beetz
Temporal projection

(a location ...
  (not-hindering (the activity (type (pick-up Cup1))))))

\[\downarrow\]

projectPlan(PickUp(Cup1), ?Tl) ∧ bagof(?F, flawsInTimeline(?F, ∅))

1. Execute plan in projection mode
2. Projection generates a timeline
3. Match pre-defined flaws on the timeline
Flip the pancake carefully

(perform (an action
  (type flip)
  (object (the object
    (type pancake)
    (on oven)))
  (params argmin_{params} P(fail(action(params),sit)
    | execute(action(params),sit),
    obj-acted-on(pancake),
    props(sit))}
Conclusions

- what is the problem? understanding by building
- take the human instructions as an indication for
  - what information is needed?
  - what not?
- shared representation
  - constraint- and optimization-based action specifications
- predictive decision making
Thank you for your attention

Questions?