Autonomous Learning for Human-scale Everyday Manipulation Tasks

Michael Beetz
University of Bremen
The Research Field:
Autonomous Robotics
Autonomous Robotics has become a Disruptive Technology

Disruptive technologies =

technologies that will transform life, business, and the global economy

source: Report from McKinsey Global Institute, May 2013

Disruptive Technologies 1-6

1. Mobile Internet
2. Automation of knowledge work
3. The Internet of Things
4. Cloud technology
5. Advanced robotics
6. (Near-)autonomous vehicles

Disruptive Technologies 7-12

7. Next-generation genomics
8. Energy storage
9. 3D printing
10. Advanced materials
11. Advanced oil/gas exploration/recovery
12. Renewable energy

Research Field

Problem

Plan Design

Learning

Conclusions

Michael Beetz

September, 2014
## Artificial Intelligence

### Disruptive Technology Systems

#### Autonomous Driving

[Google]

#### Watson

[IBM]

#### Siri Agent

[Siri/Apple]

- **Google glasses**: knowing everything about what you see (Google goggles)
- **Interactive virtual reality games**: Oculus Rift, Kinect 2, Leap motion sensor, etc
- ...
The Google Disruptive Technology Robot

- **application???
  - warehouse robot???, picking items on an order list and loading them in packages???
  - delivery robot???, delivering items to people's homes

- **expected capability???
  - capable perception-guided autonomous manipulation
  - longterm autonomy
The Research Problem:
Human-scale Manipulation Tasks

If you know the solution before understanding the problem you can be sure to be wrong.

Drew McDermott

The formulation of a problem is often more essential than its solution.

Albert Einstein

A problem well put is half solved.

John Dewey

If I had an hour to solve a problem, I'd spend 55 min thinking about the problem and 5 minutes thinking about solutions.

Albert Einstein
Autonomous Robotic Agents
Where we are, where we are going

PERFORMING A TASK
Autonomous Robotic Agents
Where we are, where we are going

PERFORMING A TASK

MASTERING A JOB

Research Field | Problem | Plan Design | Learning | Conclusions
---|---|---|---|---

Michael Beetz
September, 2014
Autonomous Robotic Agents
Where we are, where we are going

PERFORMING A TASK

MASTERING A JOB

Research Field  Problem  Plan Design  Learning  Conclusions

Michael Beetz
September, 2014
The Holy Grail: Goal-directed Object Manipulation

evolution of cognitive capabilities:

- representation
- language
- cultural learning
Artificial Intelligence

Classical AI Answer

given:

**initial state**
- ingredients
- & tools

**goal:**
- have(pancakes)
Artificial Intelligence

Classical AI Answer

**given:**
- **initial state**
  - ingredients
  - & tools

- **goal:**
  - have(pancakes)

**compute plan:**
- pour (pancakemix, bottle, oven)
- wait (3 min)
- flip (pancake)
- wait (3 min)
- put (pancake, plate)

---

Research Field | Problem | Plan Design | Learning | Conclusions
--- | --- | --- | --- | ---
Michael Beetz | | | | September, 2014

**BAYC0GROB**

13
Artificial Intelligence

Classical AI Answer

given:

initial state

ingredients & tools

have(pancakes)

compute plan:

pour (pancakemix, bottle, oven)

wait (3min)

flip (pancake)

wait (3min)

put (pancake, plate)

intelligence lies in the sequencing of actions

Research Field

Problem

Plan Design

Learning

Conclusions

Michael Beetz

September, 2014

BAYCOGROB

14
Instructions and Actions

**tools:**
- frying pan
- spatula

**ingredients:**
- pancake mix
- milk

**Steps:**
- Take the mix from the refrigerator.
- Add 400ml of milk; shake the bottle head down for 1 Minute. Let the pancake-mix sit for 2-3 minutes, shake again.
- Pour the mix into the frying pan.
- Wait for 3 minutes.
- Flip the pancake around.
- Wait for 3 minutes.
- Place the pancake onto a plate.
## Instructions and Actions

### tools:
- frying pan
- spatula

### ingredients:
- pancake mix
- milk

### Steps:
1. Take the mix from the refrigerator.
2. Add 400ml of milk; shake the bottle head down for 1 minute. Let the pancake-mix sit for 2-3 minutes, shake again.
3. Pour the mix into the frying pan.
4. Wait for 3 minutes.
5. Flip the pancake around.
6. Wait for 3 minutes.
7. Place the pancake onto a plate.

---

Michael Beetz  
September, 2014
Key Concept: Action Descriptions

*Action descriptions* represent attributes of actions, objects, etc. that are expected to be important for the skillful execution of actions.

- pour stuff from pot
- grasp the pot by the handles
- hold the pot horizontally
- tilt the pot around the axis between the handles
- hold the lid while pouring
- etc

Descriptions can be incomplete, ambiguous, inaccurate, and inconsistent.
Mastering Everyday Manipulation is Knowledge-intensive!

information in parameterized plans
- vague instruction (eg, set table, clean up)

= knowledge required by robotic agents
How much Knowledge Does a Robotic Agent Need?
Knowledge for Mastering Pancake Making

Making a Pancake
A robot pours a ready-made pancake mix onto a preheated pancake maker. Properly performed, the mix is poured onto the center of the pancake maker without spilling where it forms a round shape. The robot lets it cook until the underside of the pancake is golden brown and its edges are dry. Then, the robot carefully pushes a spatula under the pancake, lifts the spatula with the pancake on top, and quickly turns its wrist to put the pancake upside down back onto the pancake maker. The robot waits for the other side of the pancake to cook fully. Finally, it places the pancake using the spatula onto an upturned dinner plate.

What happens if: the robot pours too much pancake mix onto the pancake maker? too little? the robot pours the mix close to the edge of the pancake maker? the robot flips the pancake too soon? too late? the robot pushes only half of the spatula’s blade under the pancake? the robot turns its wrist too slow? the robot uses a knife/fork/spoon to flip the pancake? the pancake mix is too thick? too thin? the ingredients of the mix are not homogeneously mixed?
Where Does the Knowledge Come From?

Everyday Activity =

- a complex task that is both common and mundane to the agent performing it;
- one about which an agent has a great deal of knowledge, which comes as a result of the activity being common, and is the primary contributor to its mundane nature; and
- one at which adequate or satisficing performance rather than expert or optimal performance is required.

adopted from [Anderson, 1995]
Everyday manipulation is really hard
Picking up an object

decide on

- where to stand?
- which hand(s) to use?
- how to reach? …
- which grasp? where?
- how much force/lift force?
- how to lift? how to hold?
Everyday manipulation is really hard
Picking up an object

deide on

- where to stand?
- which hand(s) to use?
- how to reach? . . .
- which grasp? where?
- how much force/lift force?
- how to lift? how to hold?

based on context:

- object, object states, environment, task, . . .
Artificial Intelligence

Everyday manipulation is really hard
Picking up an object

decide on

- where to stand?
- which hand(s) to use?
- how to reach? . . .
- which grasp? where?
- how much force/lift force?
- how to lift? how to hold?

based on context:

- object, object states, environment, task, . . .

Challenge

- doing the *appropriate* thing
- to the *appropriate* object
- in the *appropriate* way
Manipulation Actions

AI vs control engineering view

**AI**: symbolic goals, qualitative relations between objects

**Control**: continuous geometric relations between coordinate frames

---

<table>
<thead>
<tr>
<th>Research Field</th>
<th>Problem</th>
<th>Plan Design</th>
<th>Learning</th>
<th>Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michael Beetz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September, 2014</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Assessment of the Research Problem

Autonomous Learning for Human-scale Manipulation Tasks

- robot control is knowledge intensive
- most knowledge/learning is needed for **how** actions are to be executed
- **hypothesis**: autonomous robot learning of human-scale manipulation tasks is not possible without robots that:
  1. know what they are doing
  2. can read, watch, and play games

- lifelong learning
- learning everything
We investigate 3 aspects

1. the memorization of execution episodes

2. the design of plans

3. the learning of structured joint probability distributions
III

Memories for Autonomous Learning:
Robots that know what they are doing
Memories for Autonomous Learning

III.1 Memories as Query Answering Systems

III.2 Learning Scenarios

III.3 “Big Data” from Manipulation Episodes

III.4 Data Sources for Manipulation Episodes

III.5 KR lite & Data Analytics

III.6 Open-EASE
Memorizing Experiences for Autonomous Learning

robots that know what they are doing

- can answer queries about
  - what they did,
  - why,
  - what happened,
  - what the effects were,
  - what they saw,
  - what they reasoned,
  - . . .
Reasoning about Specific Situations
“Is an object in the assumed field of view?”

Yields explanations for:
- objects not found
- objects possibly occluded

?- task_start(log: 'CRAMPerceive_uocvmivw', _St),
   owl_individual_of(pr2:pr2_head_mount_kinect_rgb_link, srdl2comp:'Camera'),
   obj_visible_in_camera(log: 'VisualPerception_Z9fXhEae_object_0',
                          pr2:pr2_head_mount_kinect_rgb_link, _St).
   true.

?- task_start(log: 'CRAMPerceive_uocvmivw', _St),
   owl_individual_of(Cam, srdl2comp:'Camera'),
   obj_visible_in_camera(log: 'VisualPerception_Z9fXhEae_object_0',
                          Cam, _St).
   Cam = pr2:pr2_high_def_frame ;
   Cam = pr2:pr2_head_mount_kinect_ir_link ;
   Cam = pr2:pr2_head_mount_kinect_rgb_link ;
   [...] false.
Artificial Intelligence

Objects occluded by robot parts

?- task_start(log: 'CRAMPPerceive_uocvmivw', _St),
   sub_component(pr2: pr2_right_arm, Part),
   obj_blocked_by_in_camera(log: 'VisualPerception_Z9fXhEae_object_0',
                            Part,
                            pr2: pr2_head_mount_kinect_rgb_link, _St).

Part = pr2: pr2_r_wrist_roll_link;
Part = pr2: pr2_r_forearm_cam_optical_frame;
Part = pr2: pr2_r_gripper_palm_link;
[...]

Research Field | Problem | Plan Design | Learning | Conclusions
--- | --- | --- | --- | ---
Michael Beetz | September, 2014 | | |
Memories for Autonomous Learning

III.1 Memories as Query Answering Systems

III.2 Learning Scenarios

III.3 “Big Data” from Manipulation Episodes

III.4 Data Sources for Manipulation Episodes

III.5 KR lite & Data Analytics

III.6 Open-EASE
Example learning Scenarios

• learning prediction models
• learning perception capabilities
• learning places from which objects can be perceive and reached
• …
Memories for Autonomous Learning

III.1 Memories as Query Answering Systems
III.2 Learning Scenarios

**III.3 “Big Data” from Manipulation Episodes**

III.4 Data Sources for Manipulation Episodes
III.5 KR lite & Data Analytics
III.6 Open-EASE
Learning Architecture

Expected Failures, Sensor Values, Time Spans

Diagnosing Predicted Behaviour

Active Parameters

Sensor Readings

Tracking Active State via Context Change

Logging { Symbolic Plan Events, Continuous Sensor Data

Modelling Internal Plan Structure

Logged Episodic Memories

Symbolic plan events

Continuous sensor data

Task-Pick-Object

With-Failure-Handling

Task-Navigate

Task-Perceive

Task-Grasp

With-Designators

With-Designators

Generalized Task Model

Compound Model

Research Field

Problem

Plan Design

Learning

Conclusions

Michael Beetz

September, 2014
Memory System Overview

Query interface for memory retrieval

KnowRob knowledge base

"Virtual knowledge base" interface

MongoDB database

Continuous sensor data

Plan events

Sensor data

Research Field | Problem | Plan Design | Learning | Conclusions
--- | --- | --- | --- | ---
Michael Beetz | September, 2014 | | | BAYCOGROB

36
Symbolic Plan Data

Task: Approach and pick up an object from the kitchen counter

(let* ((loc-desig (a location '((on Cupboard) (name kitchen_island)))))
  (obj-desig (an object '((type container) (at ,loc-desig))))
  (achieve '(object-in-hand ,obj-desig))))
Symbolic Plan Data

Task: Approach and pick up an object from the kitchen counter

(\texttt{let* ((loc-desig (a location \textquoteleft\textquoteleft(on Cupboard) (name kitchen_island)))))
(obj-desig (an object \textquoteleft\textquoteleft(type container) (at ,loc-desig))))
(achieve \textquoteleft\textquoteleft(object-in-hand ,obj-desig))))
Symbolic Plan Data

Task: Approach and pick up an object from the kitchen counter

(let* ((loc-desig (a location '((on Cupboard) (name kitchen_island)))))
  (obj-desig (an object '((type container) (at ,loc-desig))))
  (achieve '(object-in-hand ,obj-desig))))
Symbolic Plan Data

Task: **Approach and pick up an object from the kitchen counter**

```
(let* (((loc-desig (a location '((on Cupboard) (name kitchen_island))))
        (obj-desig (an object '((type container) (at ,loc-desig))))
        (achieve '(object-in-hand ,obj-desig))))
```
Symbolic Plan Data

Task: Approach and pick up an object from the kitchen counter

(let* (((loc-design (a location '((on Cupboard) (name kitchen_island)))))
       (obj-design (an object '((type container) (at ,loc-design))))
       (achieve '(object-in-hand ,obj-design))))
Symbolic Plan Data

Task: Approach and pick up an object from the kitchen counter

Some Object

<table>
<thead>
<tr>
<th>TYPE</th>
<th>CONTAINER</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>OBJECT</td>
</tr>
<tr>
<td>frame-id</td>
<td>&quot;head_mount_kinect_rgb_optical_frame&quot;</td>
</tr>
</tbody>
</table>

POSE

<table>
<thead>
<tr>
<th>position</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
</tr>
<tr>
<td>y</td>
</tr>
<tr>
<td>z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
</tr>
<tr>
<td>y</td>
</tr>
<tr>
<td>z</td>
</tr>
<tr>
<td>w</td>
</tr>
</tbody>
</table>

( let* ((loc-desig (a location (((on Cupboard) (name kitchen_island))))))
  (obj-desig (an object (((type container) (at ,loc-desig)))))
  (achieve '(object-in-hand ,obj-desig))))
Symbolic Plan Data

Task: Approach and pick up an object from the kitchen counter

Research Field Problem Plan Design Learning Conclusions
Michael Beetz September, 2014
Memories for Autonomous Learning

III.1 Memories as Query Answering Systems
III.2 Learning Scenarios
III.3 “Big Data” from Manipulation Episodes

**III.4 Data Sources for Manipulation Episodes**

III.5 KR lite & Data Analytics
III.6 Open-EASE
Memories for Autonomous Learning

III.1 Memories as Query Answering Systems
III.2 Learning Scenarios
III.3 “Big Data” from Manipulation Episodes
III.4 Data Sources for Manipulation Episodes
III.5 KR lite & Data Analytics
III.6 Open-EASE
OpenEASE Architecture

Research Field | Problem | Plan Design | Learning | Conclusions
--- | --- | --- | --- | ---
Michael Beetz |  |  |  | BAYC0GROB
September, 2014 |  |  |  | 46
## Predicates on Experiences

<table>
<thead>
<tr>
<th>Meta-Predicates (belief state or ground truth)</th>
<th>Reasoning about events</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>holds(occ, T_i)</code></td>
<td><code>loc_change(Obj)</code></td>
</tr>
<tr>
<td><code>belief_at(event, T_i)</code></td>
<td><code>object_perceived(Obj)</code></td>
</tr>
<tr>
<td><code>occurs(event, T_i)</code></td>
<td><strong>Reasoning about occasions</strong></td>
</tr>
<tr>
<td></td>
<td><code>loc(obj, Loc)</code></td>
</tr>
<tr>
<td></td>
<td><code>object_visible(Obj)</code></td>
</tr>
<tr>
<td></td>
<td><code>object_placed_at(Obj, loc)</code></td>
</tr>
</tbody>
</table>

**Reasoning about the logged task tree**

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>task(Task)</code></td>
<td>Tasks on interpretation stack</td>
</tr>
<tr>
<td><code>task_class(Task, Class)</code></td>
<td>Class of task</td>
</tr>
<tr>
<td><code>task_goal(Task, Goal)</code></td>
<td>Goal of task</td>
</tr>
<tr>
<td><code>task_start(Task, T)</code></td>
<td>Start time of task</td>
</tr>
<tr>
<td><code>task_end(Task, T)</code></td>
<td>End time of task</td>
</tr>
<tr>
<td><code>task_status(Task, Status)</code></td>
<td>Status of task (not started, ongoing or finalized)</td>
</tr>
<tr>
<td><code>subtask(Task, Subtask)</code></td>
<td>Task is a parent of Subtask</td>
</tr>
<tr>
<td><code>subtask^+(Task, Subtask)</code></td>
<td>Task is an ancestor of Subtask</td>
</tr>
<tr>
<td><code>returned_value(Task, Result)</code></td>
<td>Result of task (success or fail)</td>
</tr>
<tr>
<td><code>failure_task(Error, Class)</code></td>
<td>Failure of task</td>
</tr>
<tr>
<td><code>failure_class(Error, Class)</code></td>
<td>Class of failures</td>
</tr>
<tr>
<td><code>failure_attribute(Err, Name, Val)</code></td>
<td>Attribute of failure</td>
</tr>
</tbody>
</table>

**Reasoning about logged poses and designators**

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>designator_type(Desig, Type)</code></td>
<td>Type of designator</td>
</tr>
<tr>
<td><code>designator_props(Desig, Prop, Val)</code></td>
<td>Property values of designator</td>
</tr>
<tr>
<td><code>obj_pose_by_desig(Obj, Pose)</code></td>
<td>Object pose from perceived designator</td>
</tr>
<tr>
<td><code>lookup_transform(Src, Tgt, T, Tf)</code></td>
<td>Logged transform from Src to Tgt at time T</td>
</tr>
<tr>
<td><code>transform_pose(P_i, Src, Tgt, T, P_o)</code></td>
<td>Transform P_i from frame Src to frame Tgt at time T</td>
</tr>
<tr>
<td><code>visible_in_cam(Obj, Cam, T)</code></td>
<td>At time T, Obj was in the field of view of Cam</td>
</tr>
<tr>
<td><code>blocked_by_in_cam(Obj, Blk, Cam, T)</code></td>
<td>At time T, Blk was blocking the view of Cam on Obj</td>
</tr>
</tbody>
</table>

---

**Research Field** | **Problem** | **Plan Design** | **Learning** | **Conclusions**
---|---|---|---|---
Michael Beetz | September, 2014 | | | BAYCOGROB

Queries based on Experiences

“Which objects were believed to be on the table?”

“Object occluded by the robot’s arm?”

“What are common failures during pick and place?”

“How probable is success of pick and place after n fails?”
Memories for Autonomous Learning

III.1 Memories as Query Answering Systems
III.2 Learning Scenarios
III.3 “Big Data” from Manipulation Episodes
III.4 Data Sources for Manipulation Episodes
III.5 KR lite & Data Analytics

III.6 Open-EASE
A Web interface to KnowRob

KnowRob Web Console

owl_subclass_of(A, knowrob:'FoodOrDrink').
owl_has(A, rdfs:label, knowrob:'Drawer').
owl_has(knowrob:'Refrigerator67', knowrob:properPhysicalParts, P).

----- more tricky queries -----
Implementation using ROS components

KnowRob

- Prolog shell
- Local visualization canvas
- Visualization marker publisher
- json_prolog ROS node

Internet

- Websocket interface

rosbridge

/visualization_marker topic
Dockerizing KnowRob

- webrob
  - Port 49153

- tenorth
  - tenorth_data
    - /home/ros/sandbox
  - Private containers per user

- beetz
  - beetz_data
    - /home/ros/sandbox

- Common, shared containers

- mongo_db
  - MongoDB

- mongo_data
  - /data/db

- knowrob_data
  - /home/ros/knowrob_data

---

Research Field | Problem | Plan Design | Learning | Conclusions
---|---|---|---|---
Michael Beetz | | | BAYCOGROB 52 | September, 2014
Plan Design
Motivation

Focus:

- human-scale everyday manipulation activities
- lifelong learning and adaptation
- “smart” and general plans

Example Plan: Fetch and Place

**Plan schema** Generalized fetch and place (partial object description)

- find an object matching the partial object description
  - go to the place where you believe the object to be
  - look for it; if necessary make it visible
- position yourself in order to grasp the object properly
- ...

An Example Plan Acquisition Episode

plan library

**Action Core**
- fluidTransfer
  - (stuff from to)
- TurningInPlace
  - (object)
- fetch
  - object
- place
  - (object destination)

**Generic Plan Schemata**
- pour
  - (stuff container destination)
- fill
  - (stuff device container)
- flip
  - (food tool)
- flip
  - (switch)

"pour the mix into the frying pan"
An Example Plan Acquisition Episode

**Action Core**
- fluidTransfer: (stuff from to)
- TurningInPlace: (object)
- fetch object
- place (object destination)

**Generic Plan Schemata**
- pour: (stuff container destination)
- fill: (stuff device container)
- flip: (food tool)
- flip: (switch)

**“pour the mix into the frying pan”**
An Example Plan Acquisition Episode

- **plan library**
- **Action Core**
  - fluidTransfer (stuff from to)
  - TurningInPlace (object)
  - fetch object
  - place (object destination)

- **Generic Plan Schemata**
  - pour (stuff container destination)
  - fill (stuff device container)
  - flip (food tool)
  - flip (switch)

- “pour the mix into the frying pan”

**Research Field**
Michael Beetz
September, 2014

**Plan Design**

- fluidTransfer
  - pour
    - pour (stuff container destination)
- TurningInPlace
  - turn
    - flip (food tool)
- fetch object
  - fetch
    - fetch (stuff)
- place (object destination)
  - place
    - place (object destination)

**Learning**

**Conclusions**
An Example Plan Acquisition Episode

Plan Acquisition Process

Action Core
- fluidTransfer (stuff from to)
- TurningInPlace (object)
- fetch object
- place (object destination)

Generic Plan Schemata
- pour (stuff container destination)
- fill (stuff device container)
- flip (food tool)
- flip (switch)

Research Field
Problem
Plan Design
Learning
Conclusions

Michael Beetz
September, 2014

BAYCORGROB
58
An Example Plan Acquisition Episode

plan library

Action Core
- fluidTransfer (stuff from to)
- TurningInPlace (object)
- fetch object
- place (object destination)

Generic Plan Schemata
- pour (stuff container destination)
- fill (stuff device container)
- flip (food tool)
- flip (switch)

Research Field | Problem | Plan Design | Learning | Conclusions
---|---|---|---|---
Michael Beetz | | | | BAYCOGROB
September, 2014 | | | | 59
An Example Plan Acquisition Episode

**Action Core**
- **fluidTransfer**
  - (stuff from to)
- **TurningInPlace**
  - (object)
- **fetch**
  - object
- **place**
  - (object destination)

**Generic Plan Schemata**
- **pour**
  - (stuff container destination)
- **fill**
  - (stuff device container)
- **flip**
  - (food tool)
- **flip**
  - (switch)

---

**Research Field**
- **Problem**
- **Plan Design**
- **Learning**
- **Conclusions**

Michael Beetz
September, 2014

BayCogRob
60
Result of the Instruction Interpretation Step

plan schema pour( st : (some stuff (type pancake-mix))
    c : (an object (type container) (contains st))
    dest : (a location (in frying pan)))
(with-roles ((stuff st)
    (container c)
    (destination dest))
(with-subactions ((take-container (an action
        (type take)
        (object-acted-on container)))
    (pouring-action (an action
        (type pouring)
        . . . )))
    . . .
    (perform take-container)
    (perform pouring-action)))
WP1, WP2, WP5 pour the pancake mix into the frying pan

**given:**

( an action  
  (type pouring)  
  (some stuff  
    (type pancake-mix))  
  (destination  
    (a location  
      (in frying pan))))

**compute:**

additional attributes that might be necessary to ensure successful execution, such as

- how the container is grasped
- how the container is held
- the pose of the container while the pouring event occurs
- the effect of pouring: amount? shape?
- movement phases and constraints
Context-directed Plan Parameterization

Put the pancake mix away

(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))))))))

Research Field

Problem

Plan Design

Learning

Conclusions

Michael Beetz
September, 2014
An Example of Context-directed Plan Parameterization

\[
\text{setof } \text{?Pose} \ \text{On}(\text{Counter, } \text{?Pose}) \ \text{?Poses} \land \text{member(?P, ?Poses)} \\
\land \ \text{Pose}(\text{Cup, } \text{?P}) \land \text{stable(\text{Cup})}
\]
An Example of Context-directed Plan Parameterization


- setof ?Pose On(Counter, ?Pose) ?Poses
- member(?P, ?Poses)
- Pose(Cup, ?P)
- stable(Cup)

Create distribution for sampling poses

<table>
<thead>
<tr>
<th>Research Field</th>
<th>Problem</th>
<th>Plan Design</th>
<th>Learning</th>
<th>Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michael Beetz</td>
<td></td>
<td></td>
<td></td>
<td>BAYCogROB</td>
</tr>
<tr>
<td>September, 2014</td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
</tbody>
</table>
An Example of Context-directed Plan Parameterization

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

- setof ?Pose On(Counter, ?Pose) ?Poses
- member(?P, ?Poses)
- Pose(Cup, ?P)
- stable(Cup)

Draw a pose sample

<table>
<thead>
<tr>
<th>Research Field</th>
<th>Problem</th>
<th>Plan Design</th>
<th>Learning</th>
<th>Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michael Beetz</td>
<td>September, 2014</td>
<td></td>
<td></td>
<td>BAYCGR0B 66</td>
</tr>
</tbody>
</table>
An Example of Context-directed Plan Parameterization

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

- setof ?Pose On(Counter, ?Pose) ?Poses
- member(?P, ?Poses)
- Pose(Cup, ?P)
- stable(Cup)

Place the mug
An Example of Context-directed Plan Parameterization


- setof ?Pose On(Counter, ?Pose) ?Poses
- member(?P, ?Poses)
- Pose(Cup, ?P)
- stable(Cup)

Simulate for 50ms, fail!
An Example of Context-directed Plan Parameterization


- setof ?Pose On(Counter, ?Pose) ?Poses
- member(?P, ?Poses)
- Pose(Cup, ?P)
- stable(Cup)

Backtrack, draw another pose sample
An Example of Context-directed Plan Parameterization


- setof ?Pose On(Counter, ?Pose) ?Poses
- member(?P, ?Poses)
- Pose(Cup, ?P)
- stable(Cup)
An Example of Context-directed Plan Parameterization

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

- setof ?Pose On(Counter, ?Pose) ?Poses
- member(?P, ?Poses)
- Pose(Cup, ?P)
- stable(Cup)

Simulate for 50ms, succeed!
Plan Structure of the Fetch Plan

?obj ← Description of an object to grasp

**Achieve in-hand (?obj)**

?locs ← mostLikelyPlaces(?obj)

For each place ?loc in ?locs until ?obj in hand

**At-Location** (a loc (to see ?obj) (at ?loc))

Perceive(?obj)

Call low-level perception routine

Object found?

y

n

Try n times at different ?loc-grasp (a loc (to grasp ?obj))

**At-Location** (?loc-grasp)

Perform (an action (to grasp) (obj-acted-on ?obj))

Object grasped?

y

n

Finish

Backtrack to new ?loc-grasp

<table>
<thead>
<tr>
<th>Research Field</th>
<th>Problem</th>
<th>Plan Design</th>
<th>Learning</th>
<th>Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michael Beetz</td>
<td>September, 2014</td>
<td>BAYCogRob 72</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Learning First-order Probabilistic Model
Answer the Research Question

Can we design plans and provide a computational infrastructure such that plans can autonomously learn a joint probability distribution \( P(I, P, A, T, E, R) \) over

- their interpretation \( I \)
- the percepts \( P \) they receive and the effects \( E \) they cause
- and the relations \( R \) between \( I, P, \) and \( E \)

Given \( P(I, P, A, T, E, R) \)

the robot can compute:

- \( P(Q \mid \text{TaskOutcome(success,T) } \land \text{Context}) \)
- \( P(E \mid P, \text{context}) \)
- \( \ldots \)
Case Study 1: Learning Models of Perceive Plans
RoboSherlock

Michael Beetz
September, 2014

Research Field
Problem
Plan Design
Learning
Conclusions

Artificial Intelligence

Case Study 1: Learning Models of Perceive Plans
RoboSherlock

Michael Beetz
September, 2014

Research Field
Problem
Plan Design
Learning
Conclusions

Artificial Intelligence

Case Study 1: Learning Models of Perceive Plans
RoboSherlock

Michael Beetz
September, 2014

Research Field
Problem
Plan Design
Learning
Conclusions

Artificial Intelligence
Case Study 1: Learning Models of Perceive Plans

1. Extract 3D Clusters → Image ROI Conversion → FlatObject Annotation → Prim. Shape Annotation → SAC Model Annotation → Color Annotation → Linemod → Goggles Annotation

   For all point clusters $c_i$: perceive all evidence $E$ from the annotators
   - color($c_1$, yellow)
   - shape($c_1$, box)
   - logo($c_1$, Dr_Oetker)
   - color($c_2$, green)
   - color($c_3$, yellow)
   - shape($c_2$, box)
   - C1
   - C2
   - C3
   - C4

   Application of Feature Annotators

2. 50 typical real world scenarios with manually labelled object categories and context information

   - scene(breakfast)
   - scene(drawer)
   - scene(fridge)
   - category($c_1$, cereals)
   - category($c_2$, bowl)
   - category($c_3$, spoon)
   - category($c_4$, fork)
   - category($c_5$, juice)
   - category($c_6$, milk)
   - category($c_7$, ketchup)

   Statistical Relational Learner

3. For each query $q_i$ in $Q$ infer $v_i$: $\arg\max_{v_i} P(q_i = v_i | E)$

   $P(\text{category, size, color, logo, text, shape, ...})$

Research Field: Michael Beetz
Problem: September, 2014
Plan Design
Learning: Conclusions

BAYCOGROB
77
Case Study 1: Learning Models of Perceive Plans

<table>
<thead>
<tr>
<th>annotator</th>
<th>annotates if</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>always</td>
<td>color(c,col)</td>
</tr>
<tr>
<td>Size</td>
<td>always</td>
<td>size(c,s)</td>
</tr>
<tr>
<td>Goggles</td>
<td>if Google goggles returns text or logos</td>
<td>logo(c,logo) text(c,text) texture(c,t)</td>
</tr>
<tr>
<td>FlatObject</td>
<td>if there are objects that are too flat to be found by the general 3D clustering</td>
<td>shape(c,flat)</td>
</tr>
<tr>
<td>PrimShape</td>
<td>always</td>
<td>shape(c,shp)</td>
</tr>
<tr>
<td>LineMod</td>
<td>confidence that c is one of the objects looked for exceeds threshold</td>
<td>identity(c,i)</td>
</tr>
<tr>
<td>SACmodel</td>
<td>if enough inliers for a model are found</td>
<td>shape(c,sac)</td>
</tr>
<tr>
<td>Location</td>
<td>always</td>
<td>scene(c,loc)</td>
</tr>
</tbody>
</table>

Tabelle: Description of the annotators how they work, and what are the resulting annotations.
## Case Study 1: Learning Models of Perceive Plans

<table>
<thead>
<tr>
<th>Prediction/Truth</th>
<th>Bowl</th>
<th>Cereal</th>
<th>Chips</th>
<th>Coffee</th>
<th>Cup</th>
<th>Fork</th>
<th>Juice</th>
<th>Ketchup</th>
<th>Knife</th>
<th>Milk</th>
<th>Mondamin</th>
<th>Oil</th>
<th>Pancake_maker</th>
<th>Pitcher</th>
<th>Plate</th>
<th>Popcorn</th>
<th>Pot</th>
<th>Salt</th>
<th>Spatula</th>
<th>Spoon</th>
<th>Toaster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bowl</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cereal</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Chips</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Coffee</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cup</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fork</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Juice</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ketchup</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Knife</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Milk</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mondamin</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Oil</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pancake_maker</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pitcher</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Plate</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Popcorn</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pot</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Salt</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Spatula</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Spoon</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>Toaster</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Research Field</th>
<th>Problem</th>
<th>Plan Design</th>
<th>Learning</th>
<th>Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michael Beetz</td>
<td>September, 2014</td>
<td>BAYCOGROB</td>
<td>79</td>
<td></td>
</tr>
</tbody>
</table>
• high categorization accuracy
• exploiting background knowledge
• exploiting co-occurrence of objects in scenes
• additional kinds of inference tasks
## Case Study 2: Learning Models for Toy Pick-and-Place

TUM-James performing a pick and place task:

<table>
<thead>
<tr>
<th>Research Field</th>
<th>Problem</th>
<th>Plan Design</th>
<th>Learning</th>
<th>Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michael Beetz</td>
<td></td>
<td></td>
<td></td>
<td>BAYCOGROB</td>
</tr>
<tr>
<td>September, 2014</td>
<td></td>
<td></td>
<td></td>
<td>81</td>
</tr>
</tbody>
</table>
Case Study 2: Learning Models for Toy Pick-and-Place

Bayesian logic network trained on execution trace data (manually defined structure):
Case Study 2: Learning Models for Toy Pick-and-Place

Query (parametrisation): Which manipulator to use, so that success is more likely for picking up an object from the front-left position?

\[
P(pick\text{Task}_\text{hand}(H) \mid task\_hasGoal(T,G) = \text{True} \land \\
\quad \text{task}\_\text{outcome}(T) = \text{SUCCEEDED} \land \\
\quad \text{hlTask}\_\text{ofTask}(H,T) = \text{True} \land \\
\quad \text{hlTaskType}(H) = \text{Pick} \land \text{pickTask}\_\text{field}(H,F) = \text{True} \land \\
\quad \text{field}\_\text{name}(F) = \text{FrontLeft})
\]

\[
\approx \langle \text{LEFT}: 0.58, \text{RIGHT}: 0.42 \rangle
\]
Case Study 2: Learning Models for Toy Pick-and-Place

**Query (prediction)**: What is the probability of being able to successfully place an object at the back-middle position with the right manipulator?

\[
P(\text{task_outcome}(T) \mid \text{task_hasGoal}(T, G) = \text{True} \land \\
\text{hlTask_ofTask}(H, T) = \text{True} \land \\
\text{hlTaskType}(H) = \text{Place} \land \\
\text{placeTask_field}(H, F) = \text{True} \land \\
\text{pickTask_hand}(H) = \text{Right} \land \\
\text{field_name}(F) = \text{BackMiddle})
\]

\[\approx \langle \text{SUCCEEDED: 0.80, FAILED: 0.20} \rangle\]
Case Study 3: PRACS — Probabilistic Action Cores
“Flip the pancake!”

Probabilistic reasoning for disambiguation and filling information gaps

\[
\text{argmax } P_{\text{Flip}}(i \in \text{c} | \text{isa}(p, \text{Pancake}), \text{ObjectActedOn}(p), \text{Instrument}(i)) = \text{Spatula}
\]
Case Study 3: PRACS — Probabilistic Action Cores

Push ActionVerb the spatula Instrument under Place the pancake Ground LocativeRelation

IntentionallyAffect is-a UsingUtensil

Holding Agent CNI (Constructional-Null-Instantiation) Hand

BodypartOfAgent

BodyPart

Entity

Hand

Handle

Spatula

FoodTurner

KitchenUtensil

UsingUtensil

Pushing

BeingLocated

Underneath

Pancake

Blade

LocativeRelation

IntentionallyAffect

UsingUtensil

BeingLocated

Underneath

Pancake

Blade

LocativeRelation

Research Field

Problem

Plan Design

Learning

Conclusions

Michael Beetz

September, 2014

BayCogRob

86
• Bayesian Cognitive Robotics = plans that learn probabilistic models of themselves
  – BayCogRob so far: understanding by building
  – tremendous potential (demonstration examples with substantial impact)
  – tip of the iceberg
• framework for Bayesian Cognitive Robotics
• longterm fetch and place (under realistic conditions)
• learning algorithms that can exploit problem structure
• incremental learning
• query and plan specialization