Cognition-enabled Everyday Manipulation and Cognitive Robot Abstract Machines

Michael Beetz
Intelligent Autonomous Systems
Technische Universität München

November 10, 2010
Dan Wolpert: motor chauvinism

Q: why do we have a brain?
A: to produce complex and adaptable movement
   ▶ movements are the only way we have to
     ▶ interact with the world
     ▶ communicate
Dan Wolpert: motor chauvinism

Q: why do we have a brain?
A: to produce complex and adaptable movement
  ▶ movements are the only way we have to
    ▶ interact with the world
    ▶ communicate
The Human Brain Is Mostly for Manipulation

Q: why do we have such a big brain?

A: to do goal-directed object manipulation

- because always doing
  - the right thing
  - to the right object
  - in the right way

is difficult
How to pick up an object?

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

- in the context of getting an object out of a kitchen container
- if the glass is filled
- in the context of using the object as a tool
- if people are present
- ...

No November 10, 2010
Two Personal Conclusions

- goal-directed object manipulation is hard!
- cognitive mechanisms including learning, reasoning and planning are needed!
what does that mean?

1. number of tasks: \( \geq 40.000 \) webpages on wikihow.com

2. tasks include tasks such as
   - clean up,
   - prepare meal,
   - building Ikea shelves,
   - repair instructions
     - underspecified
     - complex
     - require competence
     - require manipulation skills
Human-scale Everyday Manipulation

Our Goal

what does that mean?

1. number of tasks: ≥ 40,000 webpages on wikihow.com

2. tasks include tasks such as ◦ clean up, ◦ prepare meal,
   ◦ building Ikea shelves, ◦ repair instructions
     ▶ underspecified
     ▶ complex
     ▶ require competence
     ▶ require manipulation skills

necessary for

▶ robots@home
▶ robots@work

November 10, 2010
Robots making pancakes
“Concepts”
Our Working Definition of Cognition

Cognition = information processing infrastructure for decision making and action parameterization that

▶ enables an agent \( agt \)
▶ to perform a set of tasks \( tsk \)
▶ better wrt performance measure \( p \)
  (typically generality, flexibility, reliability, performance, ...)
▶ based on
  ▶ experience and learning
  ▶ knowledge/models and reasoning
  ▶ forward models and planning/prediction

about the consequences of actions
Q. How do we know that our robot is “cognitive”?

If the cognitive mechanisms (learning, reasoning, planning) enable the robot to improve its performance in terms of (◦) generality, (◦) expected utility, (◦) flexibility, and (◦) reliability.

Example: getting objects out of kitchen containers
Getting objects out of any kitchen container
## Dimensions of Cognitive Control

### Environment and task adaptation

<table>
<thead>
<tr>
<th>General Planning-based Method</th>
<th>Specialized Learned Stereotypical Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>closed loop</td>
<td>open loop</td>
</tr>
</tbody>
</table>

- **General Planning-based Method**
  - Closed loop

- **Specialized Learned Stereotypical Skills**
  - Open loop

November 10, 2010

CRAM Michael Beetz
Predictive Decision Making

Dimensions of Cognitive Control

Without Foresight

objects out of reach

With Foresight

within reach
“more knowledge means less search”

- **task:** get the pancake mix!
  - how does it look?
  - where could it be?
  - how do I handle it?

- what do I do with the thing that I am currently seeing in order to clean up?
  - what is it?
  - what state is it in?
  - where does it belong? (in general, in this environment, in this state)
  - how do I handle it?

Knowledge-enabled Control
Robots that know what they are doing...

...can...

- answer queries about
  - what they do
  - what they have done
  - how and
  - why

- ...and use this knowledge to
  - deal with execution problems
  - learn faster
  - act more reliably
  - help programmers to debug
Cognitive Robot Abstract Machine

The Interface Layer for Cognitive Robotics
adapted from Pedro Domingos: “What’s Missing in AI: the Interface Layer”

<table>
<thead>
<tr>
<th>Field</th>
<th>Interface Layer</th>
<th>Below the Layer</th>
<th>Above the Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Systems</td>
<td>virtual machines</td>
<td>hardware</td>
<td>software</td>
</tr>
<tr>
<td>Programming systems</td>
<td>high-level languages</td>
<td>compilers, optimizers, ...</td>
<td>programming</td>
</tr>
<tr>
<td>Databases</td>
<td>relational model</td>
<td>query optimization, db design, transaction mgmt</td>
<td>enterprise applications</td>
</tr>
</tbody>
</table>
adapted from Pedro Domingos: “What’s Missing in AI: the Interface Layer”

<table>
<thead>
<tr>
<th>Field</th>
<th>Interface Layer</th>
<th>Below the Layer</th>
<th>Above the Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Systems</td>
<td>virtual machines</td>
<td>hardware</td>
<td>software</td>
</tr>
<tr>
<td>Programming systems</td>
<td>high-level languages</td>
<td>compilers, optimizers, ...</td>
<td>programming</td>
</tr>
<tr>
<td>Databases</td>
<td>relational model</td>
<td>query optimization, db design, transaction mgmt</td>
<td>enterprise applications</td>
</tr>
<tr>
<td>Personal robotics</td>
<td><strong>CRAM</strong></td>
<td>grounding in robot, AI tools, the nuts and bolts of intelligent robotics, ...</td>
<td>robot application programming</td>
</tr>
</tbody>
</table>

raise the conceptual level at which service and personal robot applications are programmed!
The Idea of Interface Layers
An Interface Layer for Cognitive Robots

Cognitive Robot Abstract Machine

- knowledge processing
- cognitive perception
- decision making

ROS Robot

- programmer designs
- implements
- cognitive architecture
- cognitive robot applications
- ...

November 10, 2010

CRAM Michael Beetz
Cognition-enabled
Perception-Action Loops
Cognition-enabled Control — the Very Idea

Example: Map Acquisition and Map-based Navigation

Model Acquisition

courtesy: Wolfram Burgard
Cognition-enabled Control — the Very Idea

Example: Map Acquisition and Map-based Navigation

Model Acquisition

Model Use

November 10, 2010

CRAM Michael Beetz
Cognition-enabled Control — the Very Idea

Example: Map Acquisition and Map-based Navigation

Model Acquisition

Model Use

Where am I?

November 10, 2010
Cognition-enabled Control — the Very Idea

Example: Map Acquisition and Map-based Navigation

Model Acquisition

Model Use

Where am I?

Where is L?

courtesy: Wolfram Burgard
Cognition-enabled Control — the Very Idea

Example: Map Acquisition and Map-based Navigation

Model Acquisition

What does the environment look like?

Model Use

Where am I?

Where is L?

How do I get there?

courtesy: Wolfram Burgard

November 10, 2010
Why Cognition-enabled Control?

General Navigation Routine

```plaintext
routine navigate ⟨tsk⟩
    in parallel do continually estimate your position
    whenever you are lost do relocalize
main process
    if reachable(dest(⟨tsk⟩))
    then nav-plan ← compute-nav-plan(curr-pos, dest(⟨tsk⟩))
       execute nav-plan
```
Why Cognition-enabled Control?

General Navigation Routine

```
routine navigate ⟨tsk⟩
    in parallel do continually estimate your position
    whenever you are lost do relocalize
main process
    if reachable(dest(⟨tsk⟩))
    then nav-plan ← compute-nav-plan(curr-pos, dest(⟨tsk⟩))
    execute nav-plan
```

Cognitive mechanisms enable us to control the robot

- reliably
- flexibly
- efficiently

in concise control programs
Outline

Perception-Guided Control Programs

Cognition-Enabled Perception-Guided Control Programs

Cognition-Enabled Perception-Guided Action Plans

November 10, 2010

CRAM

Michael Beetz
Perception-Guided Control Programs
 Programs/Plans for Everyday Manipulation

- Many potential sources of error!
- Control program must detect and recover from failure cases \(\geq 90\%\) of the code
Many potential sources of error!
Control program must detect and recover from failure cases (≥ 90% of the code)
Programs/Plans for Everyday Manipulation

(EXPANDED-GOAL
 (:ACHIEVE (ENTITY-PICKED-UP ENTITY) :PURPOSE PURPOSE :SIDE (OR SIDE (USED-ARM-WITH-GOAL LOCATION)) :ACHIEVE-ENTITY-PICKED-UP
 ((ENTITY TR-RULE-NAME POSE-FAILURE-TO TOLERANCE POSE-TRIES GRIP-TRIES CARRY-TRIES SIDE PURPOSE
  (ENTITY :ACHIEVE-ENTITY-PICKED-UP (ST-CREATE :DIST 0.2 :AZ 0.3926991) 3 3 0 (OR SIDE NIL)
  (LET ((INNER-CONTACTS NIL))
   (WITH-FAILURE-HANDLING FAILURE ((CARRY-TRIES-COUNT CARRY-TRIES) (GRIP-TRIES-COUNT GRIP-TRIES))
    (RECOVER ((TYPEP FAILURE 'ENTITY-LOST-FAILURE)
               (LET ((SIDE (ENTITY-GRIPPING-SIDE ENTITY NIL)))
                (HANDLE-PLAN-FAILURE CARRY-TRIES-COUNT :ENTITY ENTITY :DO-ALWAYS ((ENTITY TR-RULE-NAME SIDE)
                 ((TYPEP FAILURE 'GRIP-FAILURE)
                  (HANDLE-PLAN-FAILURE GRIP-TRIES-COUNT :ENTITY ENTITY :DO-RETRY ((RECOVER-ARM-LOC)
                   (T (HANDLE-PLAN-FAILURE 0 :ENTITY ENTITY)))))

  (MONITOR)
  (PERFORM
   (:TAG FIND-ENTITY
    (SETF ENTITY
     (EXPANDED-GOAL (:PERCEIVE ENTITY) :PERCEIVE ((DESIGNATOR TR-RULE-NAME SKIP-TO)
      (LET* ((#:GOAL1359 (MAKE-INSTANCE 'ENTITY-FOUND :DESIGNATOR DESIGNATOR))
         (#:ROUTINE1360 (ARBITRATION #:GOAL1359 (COGITO::FILTER-SETTINGS (LIST)
          (#:ROUTINE-RES1361 NIL)))
         (SETGV :GOAL-TASK (TYPE-OF #:GOAL1359) #:TAG-GOAL1363)
         (PULSE (GETGV :GOAL-START-FLUENT (TYPE-OF #:GOAL1359)))
         (TAG #:TAG-GOAL1363)

November 10, 2010

CRAM
Michael Beetz
Interesting Numbers

- 2 activities
- 7 manipulation plans
- hierarchy of both activities is 4–7 levels deep
- six kinds of failures are monitored
- expanded plan has approximately 1200 lines
- approx. 700 conditions are tested during one run
Cognition-Enabled Perception-Guided Control Programs
instead of prespecifying decisions

\[(\text{at-location } \langle \text{OBJ.POS.x - 60, OBJ.POS.y - 10} \rangle \text{ (pick-up OBJ))})\]

let the robot infer the decision

\[(\text{at-location (the ARPlace (task (a task (task-action pick-up) (objectActedOn (a cup on table)))))})\]

\[(\text{with parameters ((reaching-trajectory ... ) (grasp-type ... )) (grasp-type ... )) (pick-up all cups))}))))\]
Cognition: Inferring Control Decisions

Lazy, evidence-based decision making

“A decision is a commitment to a plan or an action parameterization based on evidence and the expected costs and benefits associated with the outcome.”

adapted from Resulaj et al, Changes of mind in decision-making
Cognition: Decisions Based on Foresight

- **Representation:**
  - Discretized space of potential manipulation places
  - Mapping to expected utilities

- **Advantages:**
  - are learned from and are grounded in observed experience
  - take state estimation uncertainties into account
  - enable least-commitment planning
  - maximize expected utility
missingObjects(Meal, Missing):-
  instanceOf(Table, 'table'),
  in(Table, Kitchen),
  primaryFunction(Table, 'HavingAMeal'),
  perceivedObjectsOnPlane(Table, Perceived),
  neededObjectsForMeal(Perceived, Needed),
  setOf(Obj, 
    (member(Obj, Needed), 
      not( member(Obj, Perceived)), 
      Missing).)

First-Order Probabilistic Reasoning

Semantic Map, Encyclopedic Knowledge

K-Copman perception server

CRAM Michael Beetz

November 10, 2010
Cognition: Acting on the Right Objects

- Similarity measures based on different sensory information
- Dealing consistently with geometric and appearance based features in a probabilistic framework
Cognition-Enabled Perception-Guided Control Plans
Declarative Goal Hierarchies

- Achieve(Loc(Obj, Loc))
  - Perceive(Loc(Obj, Loc))
  - Achieve(ObjPlacedAt(Obj, Loc))
  - Achieve(ObjInHand(Obj))

November 10, 2010

CRAM

Michael Beetz
How do robots know what they are doing?

Robot behavior

Execution trace log file

Robot control program including plan

- sensor data
- robot state
- program state
- actions

Robot behavior generates behavior

logs

prolog

Analysis and query system

generates
behavior

(def-top-level-plan ehow-make-pancakes15 ()
  (with-designators ( mixforbakedgoods21 (object ' (type pan
    pancake23 (object ' (type pancake))))
    refrigerator23 (object ' (type refrigerator)
    (cookingvessel23 (object ' (type pan))))
    dinnerplate29 (object ' (type plate)))
  (location1 (location ' (in refrigerator23 loc
    location2 (location ' (in ,cookingvessel23
    location0 (location ' (on dinnerplate29 t

(achieve (object-in-hand ?_ ?_))
(achieve (visualize-robot-trajectory)
(achieve (object-placed-at ?_ ?_))

November 10, 2010

CRAM Michael Beetz
Plan Execution

Plan Execution

Task Tree

Belief state

Sensors

Interpreter

Plan

Actuators

Logging

November 10, 2010

CRAM Michael Beetz
Recording Execution Traces

Achieve(Loc(bottle, table))

Achieve(ObjectOpened(fridge))  Achieve(ObjPlacedAt(bottle, table))

Achieve(ObjInHand(bottle))

Achieve(ObjectClosed(fridge))
**Achieve(Loc(bottle, table))**

- **Achieve(ObjectOpened(fridge))**
- **Achieve(ObjInHand(bottle))**
- **Achieve(ObjectClosed(fridge))**

**Action:**
- Move to fridge

**Log:**
- Achieve(Loc(bottle, table)) running
- Achieve(Loc(Robot, l)) running
- Trajectory of robot
- ...

No\vember 10, 2010

CRAM Michael Beetz
Recording Execution Traces

Achieve(Loc(bottle, table))

- Achieve(ObjectOpened(fridge))
- Achieve(ObjPlacedAt(bottle, table))
- Achieve(ObjInHand(bottle))
- Achieve(ObjectClosed(fridge))

Action:
- Open fridge

Log:
- Achieve(Loc(Robot, l)) succeeded
- Achieve(ObjectOpened(fridge)) running
- Trajectory of arm
- ...

November 10, 2010
Recording Execution Traces

Achieve(Loc(bottle, table))

Achieve(ObjectOpened(fridge))
Achieve(ObjPlacedAt(bottle, table))

Achieve(ObjInHand(bottle))
Achieve(ObjectClosed(fridge))

Action:
- Grasp the bottle

Log:
- Achieve(ObjectOpened(fridge)) succeeded
- Achieve(ObjInHand(bottle)) running
- Perceived properties of bottle (object designator)
- ...
Recording Execution Traces

\[ \text{Achieve(} \text{Loc(} \text{bottle, table})\text{)} \]

\[ \text{Achieve(} \text{ObjectOpened(} \text{fridge})\text{)} \]

\[ \text{Achieve(} \text{ObjPlacedAt(} \text{bottle, table})\text{)} \]

\[ \text{Achieve(} \text{ObjInHand(} \text{bottle})\text{)} \]

\[ \text{Achieve(} \text{ObjectClosed(} \text{fridge})\text{)} \]

Action:
- Close the fridge

Log:
- …
Achieve(Loc(bottle, table))

Achieve(ObjectOpened(fridge))

Achieve(ObjPlacedAt(bottle, table))

Achieve(ObjInHand(bottle))

Achieve(ObjectClosed(fridge))

---

Action:
- Put down bottle

Log:
- ...
Recording Execution Traces

Achieve(Loc(bottle, table))

- Achieve(ObjectOpened(fridge))
- Achieve(ObjInHand(bottle))
- Achieve(ObjectClosed(fridge))
- Achieve(ObjPlacedAt(bottle, table))

Action: Move to fridge
Log:
- Achieve(Loc(bottle, table)) running
- Achieve(Loc(Robot, l)) running
- Trajectory of robot

Action: Open fridge
Log:
- Achieve(Loc(Robot, l)) succeeded
- Achieve(ObjectOpened(fridge)) running
- Trajectory of arm

Action: Grasp the bottle
Log:
- Achieve(ObjectOpened(fridge)) succeeded
- Achieve(ObjInHand(bottle)) running
- Perceived properties of bottle

Action: Close the fridge
Action: Put down bottle

November 10, 2010
Reasoning based on Execution Traces

▶ Where did you stand?
▶ How did you move?
▶ How did you move the arm while grasping the bottle?

'Prolog' query

(and (task ?tsk)
  (task-goal ?tsk
    (achieve (arms-at ?traj)))
  (task-outcome ?tsk ?outcome)
  (desig-prop ?traj (to pick-up))
  (visualize-trajectory ?tsk "l_gripper_tool_)))
Bayesian Cognitive Robotics

Perception:
- noise
- low reliability
- partial observability

Action:
- noise
- uncertain effects
- failure

Environment:
- world, action and sensor models

Task:
- ambiguity
- incomplete specification

November 10, 2010
Cognition: Learning from Execution Traces

- generate probabilistic model structures from semantic plans
- models of continuous & discrete behaviour
- learn model parameters from execution traces
- complex situational dependencies (relational descriptions)
Cognition: Reasoning Patterns

▶ Prediction

\[ P(\text{successful}(\text{Robot, Grasp, Obj, Sit}) \mid \text{graspType}(\text{Grasp, SidewaysRight}) \land \text{objectType}(\text{Obj, Cup}) \land \text{relOrientation}(\text{Robot, Cup, 0.05, Sit}) \land \text{relPos}(\text{Robot, Obj, 5.8, -3.2, Sit}) \land \text{obstructs}(\text{Clutter1, Obj, Sit}) \land \text{relPos}(\text{Clutter1, Obj, 3.45, 5.23, Sit}) \land \text{size}(\text{Clutter1, 4.2, 3.5, Sit})) \]

\[ P(\text{successful}(\text{Robot, Grasp2, Obj2, Sit2}) \mid \text{successful}(\text{Robot, Grasp1, Obj1, Sit1}) \land \text{precedes}(\text{Sit1, Sit2})) \]

▶ Evaluating Alternatives

\[ P(\text{graspType}(\text{Grasp, ?type}) \mid \text{successful}(\text{Robot, Grasp, Obj, Sit}) \land \ldots) \]

▶ Diagnosis

\[ P(\text{localizationQuality}(\text{Robot, Bad, Sit}) \mid \neg \text{successful}(\text{Robot, Grasp, Obj, Sit}) \land \ldots) \]

\[ P(\text{perceptionAccuracy}(\text{Robot, Bad, Sit}) \mid \neg \text{successful}(\text{Robot, Grasp, Obj, Sit}) \land \ldots) \]
Conclusions

Cognition-enabled Perception-Action Loops

- **Perception-guided control programs** define how a robot is to respond to sensory inputs and failures in order to accomplish its goals.

- They become **cognitive** by reasoning about control decisions in order to achieve superior...
  - robustness
  - flexibility
  - efficiency

- By turning control programs into **semantically interpretable action plans**, a robot can...
  - explicitly represent its goals and monitor success during temporal projections
  - reason about plan execution and explain its behaviour to humans
  - learn models based on data gathered during plan execution
Thanks!

Available in TUM ROS Package Repository:

http://tum-ros-pkg.svn.sourceforge.net/