Cognition-enabled Everyday Manipulation and Cognitive Robot Abstract Machines

Michael Beetz

Intelligent Autonomous Systems
Technische Universität München

November 10, 2010

The Brain, Movement, and Manipulation

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 Brain, Movement, and Manipulation

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

Even the biggest Trees Don't Have Brains

The Brain, Movement, and Manipulation (2)

Homunculus

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

Decisions, Decisions, Decisions

Goal-directed Object Manipulation

How to pick up an object?

decide on
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
- ...

Two Personal Conclusions

- goal-directed object manipulation is hard!
- cognitive mechanisms including learning, reasoning and planning are needed!

Human-scale Everyday Manipulation

Our Goal

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

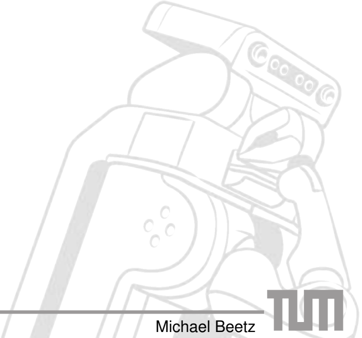
Our Goal

what does that mean?

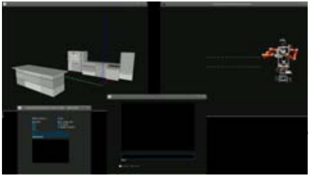
1. number of tasks: $\geq$ 40.000 webpages on wikihow.com
2. tasks include tasks such as ◦ clean up, ◦ prepare meal, ◦ building Ikea shelves, ◦ repair instructions
   - underspecified
   - complex
   - require competence
   - require manipulation skills

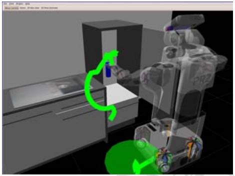
necessary for

- robots@home
- robots@work



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

Generality

Dimensions of Cognitive Control

getting objects out of any kitchen container

Getting Better

Dimensions of Cognitive Control

Environment and task adaptation

General Planning-based Method	Specialized Learned Stereotypical Skills
closed loop	open loop

Predictive Decision Making

Dimensions of Cognitive Control

Without Foresight	With Foresight
objects out of reach	within reach
	

Using Knowledge

Dimensions of Cognitive Control

"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

What's Missing in CR: The Interface Layer

... as in many other Fields

adapted from Pedro Domingos: "What's Missing in AI: the Interface Layer"

Field	Interface Layer	Below the Layer	Above the Layer
Operating Systems	virtual machines	hardware	software
Programming systems	high-level languages	compilers, optimizers, ...	programming
Databases	relational model	query optimization, db design, transaction mgmt	enterprise applications

What's Missing in CR: The Interface Layer

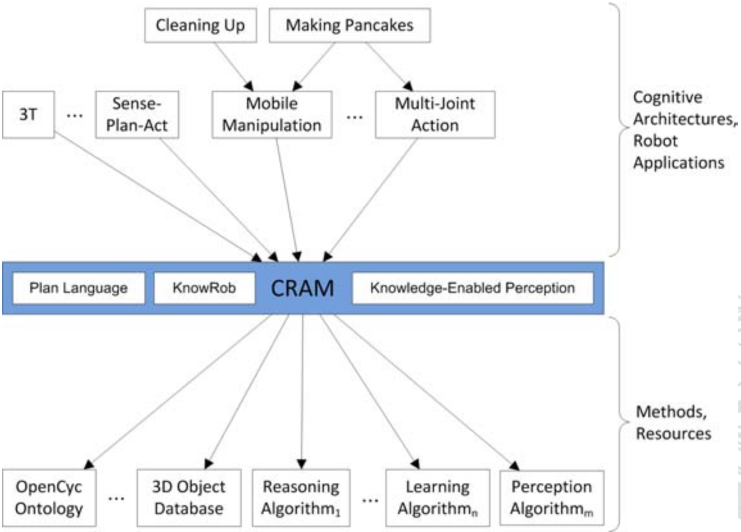
... as in many other Fields

adapted from Pedro Domingos: "What's Missing in AI: the Interface Layer"

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

raise the conceptual level at which service and personal robot applications are programmed!

The Idea of Interface Layers



An Interface Layer for Cognitive Robots

Programmer
- designs
- implements

- cognitive architecture
- cognitive robot applications
- ...

CRAM

Cognitive Robot Abstract Machine

knowledge processing	**cognitive perception**	**decision making**

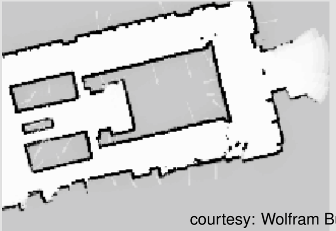
ROS Robot

Cognition-enabled Perception-Action Loops

Cognition-enabled Control — the Very Idea

Example: Map Acquisition and Map-based Navigation

Model Acquisition

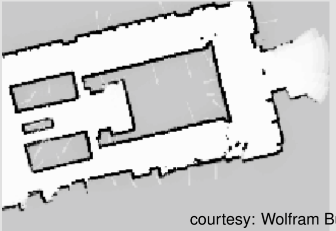


courtesy: Wolfram Burgard

Cognition-enabled Control — the Very Idea

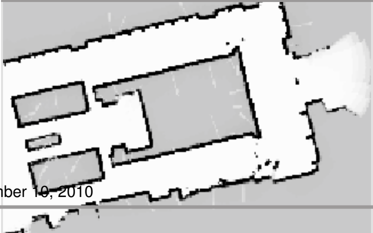
Example: Map Acquisition and Map-based Navigation

Model Acquisition



courtesy: Wolfram Burgard

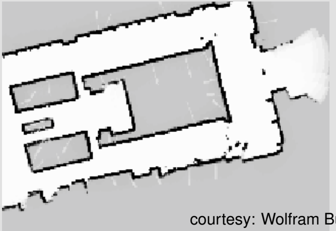
Model Use



Cognition-enabled Control — the Very Idea

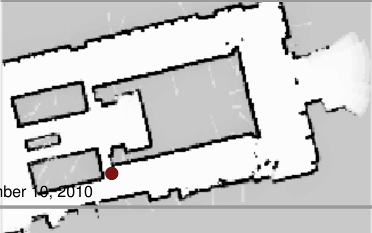
Example: Map Acquisition and Map-based Navigation

Model Acquisition



courtesy: Wolfram Burgard

Model Use

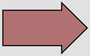


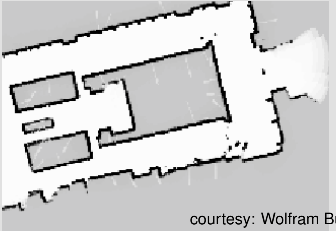
Where am I?

Cognition-enabled Control — the Very Idea

Example: Map Acquisition and Map-based Navigation

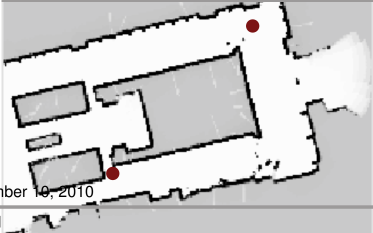
Model Acquisition





courtesy: Wolfram Burgard

Model Use



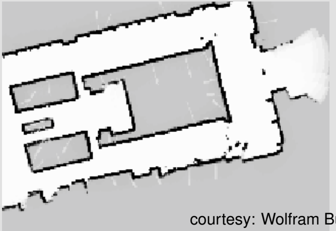
Where am I?

Where is L?

Cognition-enabled Control — the Very Idea

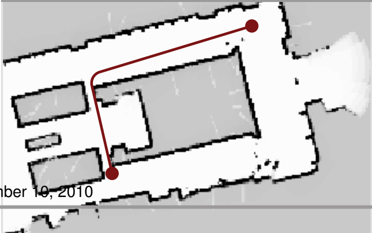
Example: Map Acquisition and Map-based Navigation

Model Acquisition



courtesy: Wolfram Burgard

Model Use



Where am I?

Where is L?

How do I get there?

Why Cognition-enabled Control?

General Navigation Routine

routine navigate $\langle tsk \rangle$
  in parallel do continually estimate your position
    whenever you are lost **do** relocalize
    main process
      if reachable(dest($\langle tsk \rangle$))
      then nav-plan $\leftarrow$ compute-nav-plan(curr-pos, dest($\langle tsk \rangle$))
        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

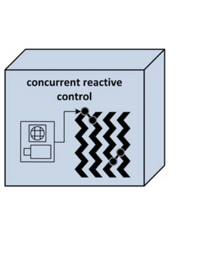
Cognitive mechanisms enable us to control the robot

- reliably
- flexibly
- efficiently

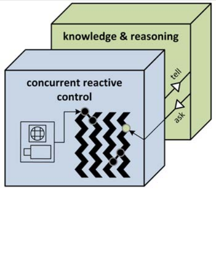
in concise control programs

Outline

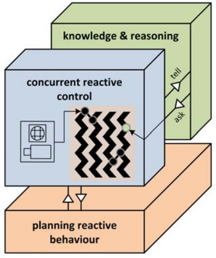
Perception-Guided Control Programs



Cognition-Enabled Perception-Guided Control Programs

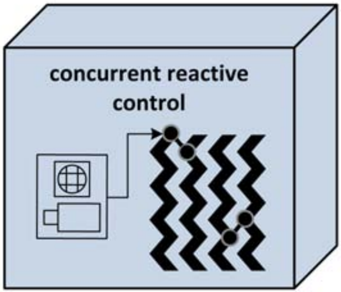


Cognition-Enabled Perception-Guided *Action Plans*

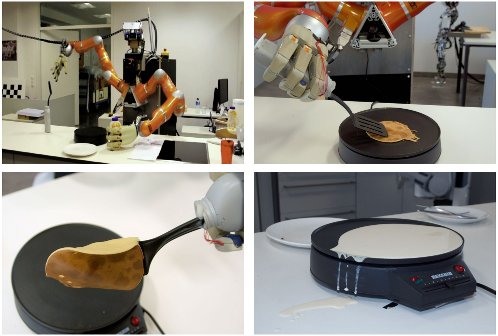


Part I

Perception-Guided Control Programs

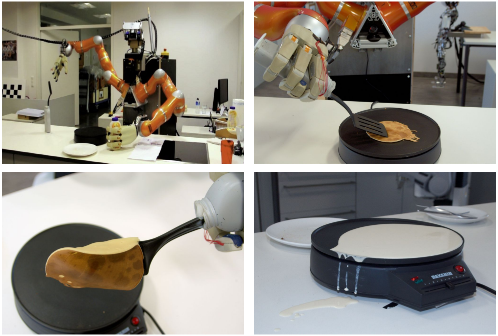


Programs/Plans for Everyday Manipulation



- Many potential sources of error!
- Control program must detect and recover from failure cases ($\geq$ 90% of the code)

Programs/Plans for Everyday Manipulation



- Many potential sources of error!
- Control program must detect and recover from failure cases ($\geq$ 90% of the code)

Programs/Plans for Everyday Manipulation

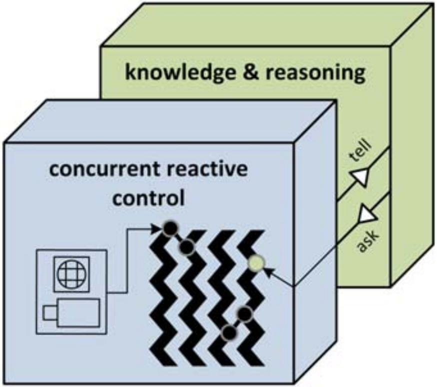
```
(EXPANDED-GOAL
 (:ACHIEVE (ENTITY-PICKED-UP ENTITY) :PURPOSE PURPOSE :SIDE (OR SIDE (USED-ARM-WITH-GOA
 :ACHIEVE-ENTITY-PICKED-UP
 ((ENTITY TR-RULE-NAME POSE-FAILURE-TOLERANCE POSE-TRIES GRIP-TRIES CARRY-TRIES SIDE PU
  (ENTITY :ACHIEVE-ENTITY-PICKED-UP (ST-CREATE :DIST 0.2 :AZ 0.3926991) 3 3 0 (OR SIDE
  NIL)
 (LET ((INNER-CONTACTS NIL))
   (WITH-FAILURE-HANDLING FAILURE ((CARRY-TRIES-COUNT CARRY-TRIES) (GRIP-TRIES-COUNT GR
    (RECOVER ((TYPEP FAILURE 'ENTITY-LOST-FAILURE)
              (LET ((SIDE (ENTITY-GRIPPING-SIDE ENTITY NIL)))
                (HANDLE-PLAN-FAILURE CARRY-TRIES-COUNT :ENTITY ENTITY :DO-ALWAYS ((ENTI
             ((TYPEP FAILURE 'GRIP-FAILURE)
              (HANDLE-PLAN-FAILURE GRIP-TRIES-COUNT :ENTITY ENTITY :DO-RETRY ((RECOVER-
             (T (HANDLE-PLAN-FAILURE 0 :ENTITY ENTITY)))
    (MONITOR)
    (PERFORM
     (:TAG FIND-ENTITY
      (SETF ENTITY
            (EXPANDED-GOAL (:PERCEIVE ENTITY) :PERCEIVE ((DESIGNATOR TR-RULE-NAME SKIP-
             (LET* ((#:GOAL1359 (MAKE-INSTANCE 'ENTITY-FOUND :DESIGNATOR DESIGNATOR))
                    (#:ROUTINE1360 (ARBITRATION #:GOAL1359 (COGITO::FILTER-SETTINGS (LI
                    (#:ROUTINE-RES1361 NIL))
               (SETGV :GOAL-TASK (TYPE-OF #:GOAL1359) #:TAG-GOAL1363)
               (PULSE (GETGV :GOAL-START-FLUENT (TYPE-OF #:GOAL1359)))
               (:TAG #:TAG-GOAL1363
```

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

Part II

Cognition-Enabled Perception-Guided Control Programs



Realization of Control Decisions

instead of prespecifying decisions

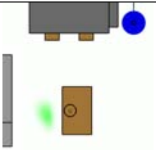
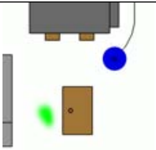
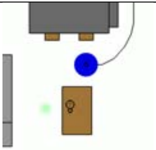
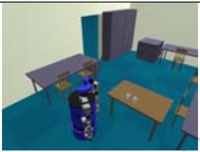
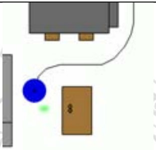
```
(at-location ⟨ OBJ.POS.x - 60, OBJ.POS.y - 10 ⟩
    (pick-up    OBJ)))))
```

let the robot infer the decision

```
(at-location (the ARPlace
               (task    (a task (task-action        pick-up)
                                (objectActedOn   (a cup     on table)))))
    (with parameters
        ((reaching-trajectory ... ) (grasp-type ... ))
         (grasp-type ... ))
           (pick-up     all cups)))))
```

Cognition: Inferring Control Decisions

Lazy, evidence-based decision making

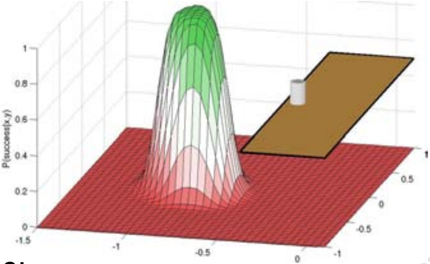
Step 1	ARPlace	Step 2	ARPlace
			
Step 3	**ARPlace**	**Step 4**	**ARPlace**
			

“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

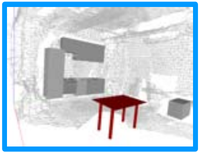
- Representation:
  - Discretized space of potential maniplation places
  - Mapping to expected utilities



- Advantages:
  - are learned from and are grounded in observed experience
  - take state estimation uncertainties into account
  - enable least-commitment planning
  - maximize expected utility

Cognition: Knowledge-Enabled Perception

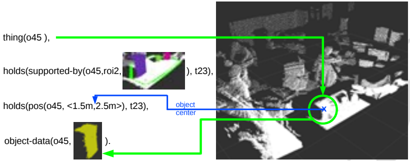
Semantic Map, Encyclopedic Knowledge



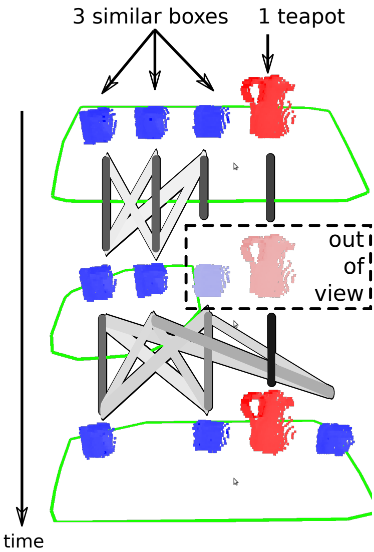
missingObjects(Meal, Missing):-
instanceOf(Table, 'table'),
in(Table, Kitchen),
primaryFunction(Table, 'HavingAMeal'),
perceivedObjectsOnPlane(Table, Perceived),
neededObjectsForMeal(Perceived, Needed),
setOf(Obj,
(member(Obj, Needed),
not(member(Obj, Perceived)),
Missing).

K-Copman perception server

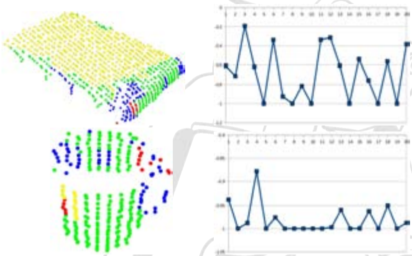
First-Order Probabilistic Reasoning



Cognition: Acting on the Right Objects

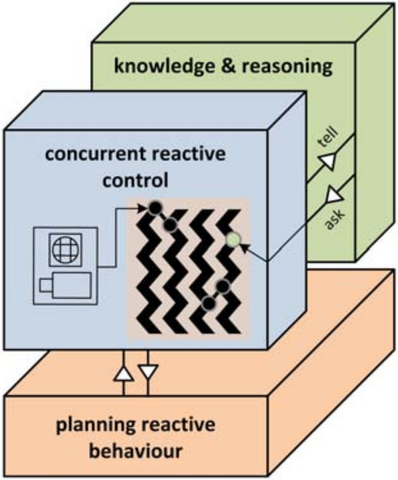


- Similarity measures based on different sensory information
- Dealing consistently with geometric and appearance based features in a probabilistic framework

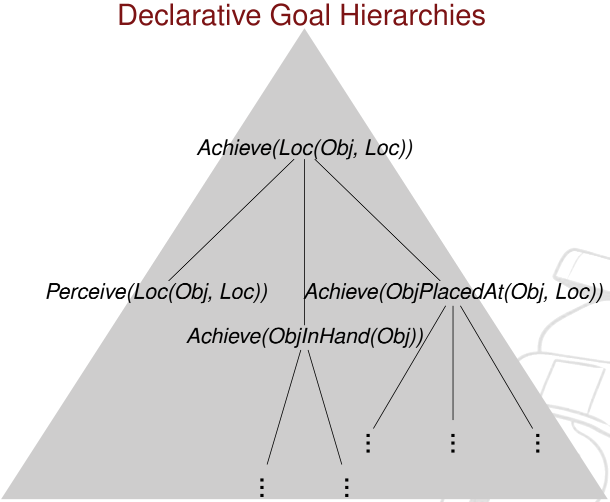


Part III

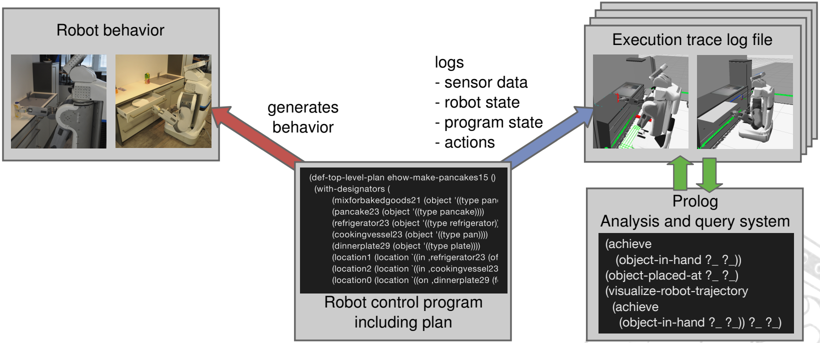
Cognition-Enabled Perception-Guided Control Plans



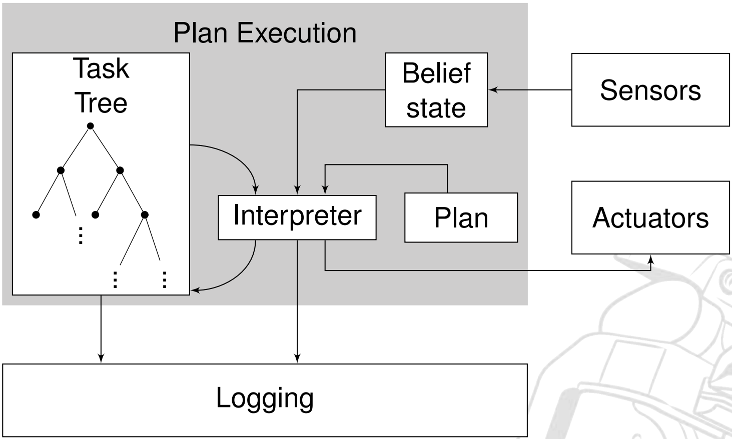
Plans


Declarative Goal Hierarchies
Achieve(Loc(Obj, Loc))
Perceive(Loc(Obj, Loc))
Achieve(ObjPlacedAt(Obj, Loc))
Achieve(ObjInHand(Obj))

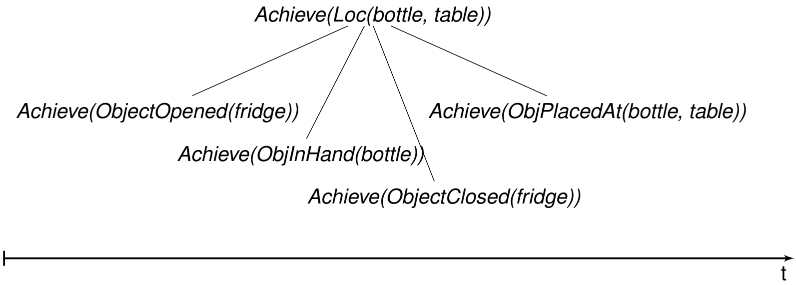
How do robots know what they are doing?



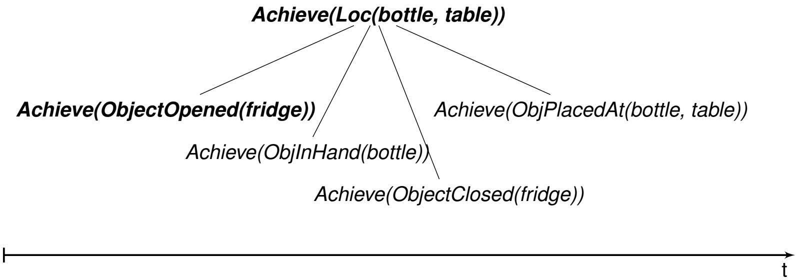
Plan Execution

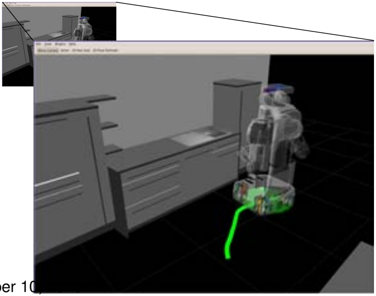


Recording Execution Traces



Recording Execution Traces





Action:

- Move to fridge

Log:

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

Recording Execution Traces

Achieve(Loc(bottle, table))

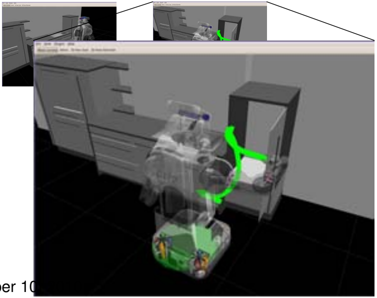
Achieve(ObjectOpened(fridge))

Achieve(ObjInHand(bottle))

Achieve(ObjectClosed(fridge))

Achieve(ObjPlacedAt(bottle, table))

t



Action:

- Open fridge

Log:

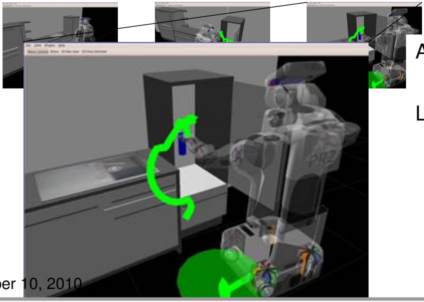
- Achieve(Loc(Robot, l)) succeeded
- Achieve(ObjectOpened(fridge)) running
- Trajectory of arm
- . . .

Recording Execution Traces

Achieve(Loc(bottle, table))

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

t



Action:

- Grasp the bottle

Log:

- Achieve(ObjectOpended(fridge)) succeeded
- Achieve(ObjInHand(bottle)) running
- Perceived properties of bottle (object designator)
- . . .

Recording Execution Traces

Achieve(Loc(bottle, table))

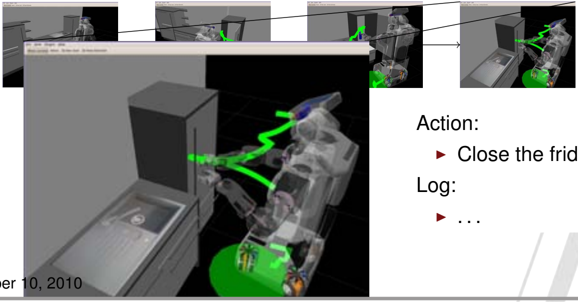
Achieve(ObjectOpened(fridge))

Achieve(ObjInHand(bottle))

Achieve(ObjectClosed(fridge))

Achieve(ObjPlacedAt(bottle, table))

t



Action:

- Close the fridge

Log:

- . . .

Recording Execution Traces

Achieve(Loc(bottle, table))

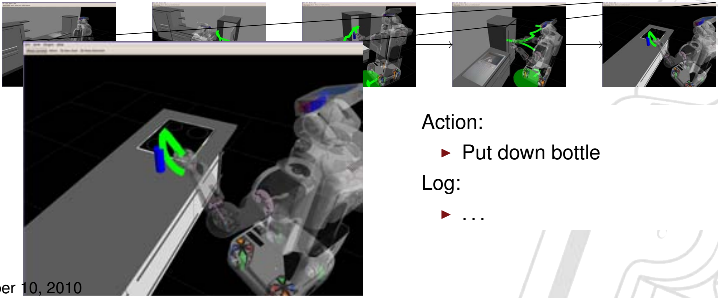
Achieve(ObjectOpened(fridge))

Achieve(ObjInHand(bottle))

Achieve(ObjectClosed(fridge))

Achieve(ObjPlacedAt(bottle, table))

t



Action:

- Put down bottle

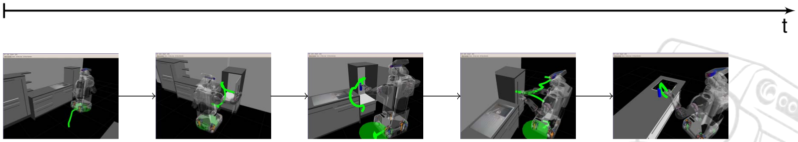
Log:

- ...

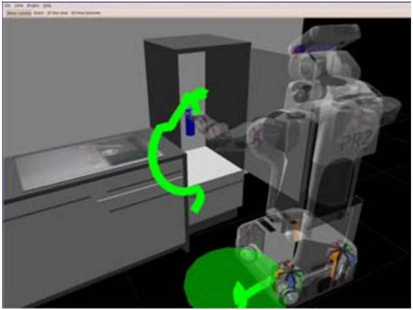
Recording Execution Traces

Achieve(Loc(bottle, table))

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



Reasoning based on Execution Traces

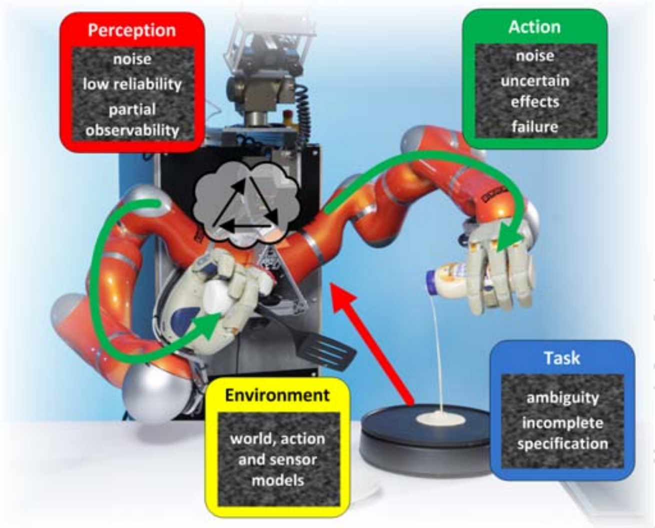


- Where did you stand?
- How did you move?
- How did you move the arm while grasping the bottle?

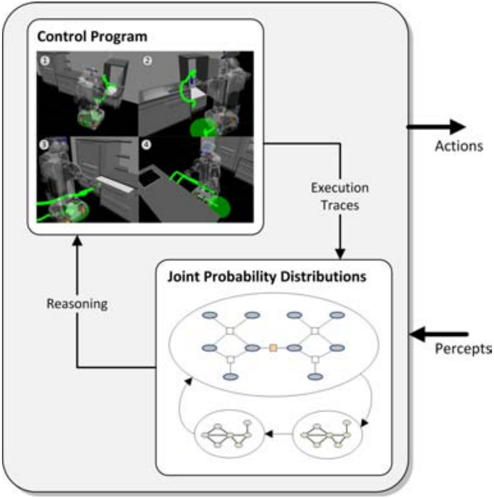
’Prolog’ query

```
(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



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(successful(Robot, Grasp, Obj, Sit) |
  graspType(Grasp, SidewaysRight) ∧ objectType(Obj, Cup) ∧
  relOrientation(Robot, Cup, 0.05, Sit) ∧ relPos(Robot, Obj, 5.8, -3.2, Sit) ∧
  obstructs(Clutter1, Obj, Sit) ∧ relPos(Clutter1, Obj, 3.45, 5.23, Sit) ∧
  size(Clutter1, 4.2, 3.5, Sit))
  P(successful(Robot, Grasp2, Obj2, Sit2) |
  successful(Robot, Grasp1, Obj1, Sit1) ∧ precedes(Sit1, Sit2))

- **Evaluating Alternatives**

  P(graspType(Grasp, ?type) |
  successful(Robot, Grasp, Obj, Sit) ∧ . . .)

- **Diagnosis**

  P(localizationQuality(Robot, Bad, Sit) |
  ¬successful(Robot, Grasp, Obj, Sit) ∧ . . .)
  P(perceptionAccuracy(Robot, Bad, Sit) |
  ¬successful(Robot, Grasp, Obj, Sit) ∧ . . .)

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

Thanks!

Available in TUM ROS Package Repository:

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