Cognition-Enabled Robot Control for the Realization of Home Chore Task Intelligence

Michael Beetz

Intelligent Autonomous Systems
Technische Universität München

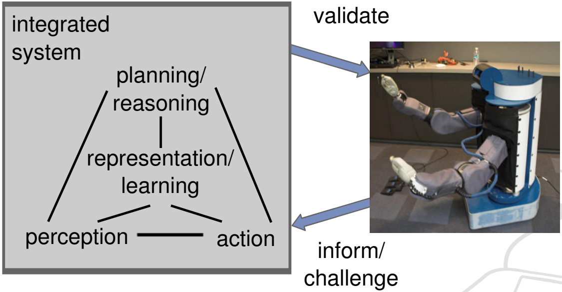
LCCC & Rosetta Symposium on
Robot Skill Learning and Cognition
April 18, 2012

Intelligent
Autonomous
Systems

Everyday Robot Manipulation

Where we want to go

Nils Nilsson's challenge: a robot that can do what is reasonable to expect from it given its sensors and actuators

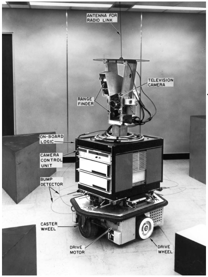


[Eric Berger, Stanford Univ]

Robotic roadmaps and white papers: robot (co-)workers, autonomous robot assistants, robot companions

Everyday Robot Manipulation

Where We are :-)



- representation & reasoning
- uncertainty
- planning
- learning
- vision

What we can be proud of . . .

Integrated Systems

Autonomous Driving

[Google]

Watson

[IBM]

Siri Agent

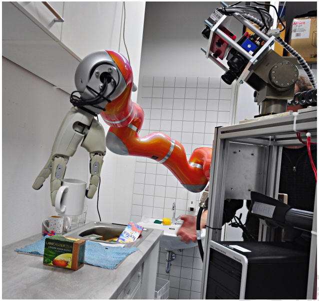
[Siri/Apple]

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 to grasp?
- how much force?
- how much lift force?
- how to lift?
- how to hold?

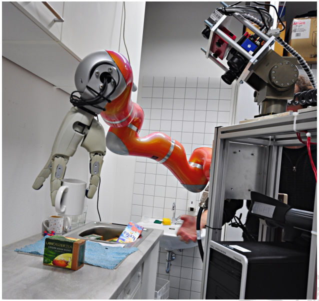


Everyday manipulation is really hard

Picking up an object

decide on

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



based on context:

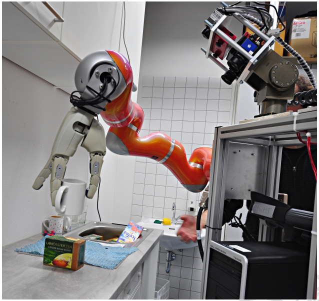
- object, object states, environment, task, . . .

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 to grasp?
- how much force?
- how much lift force?
- how to lift?
- how to hold?



based on context:

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

Challenge

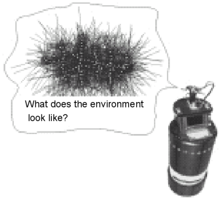
- doing the appropriate thing
- to the appropriate object
- in the appropriate way

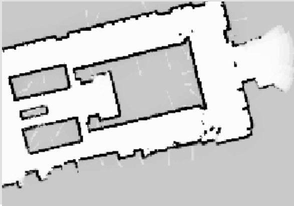
Cognition-enabled Control

Cognition-enabled Control — the Very Idea

Example: Map Acquisition and Map-based Navigation

Model Acquisition



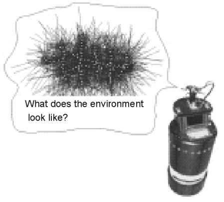


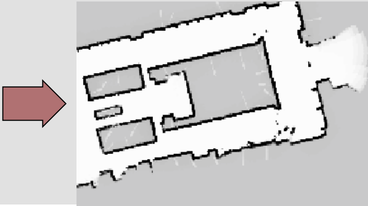
courtesy: Wolfram Burgard

Cognition-enabled Control — the Very Idea

Example: Map Acquisition and Map-based Navigation

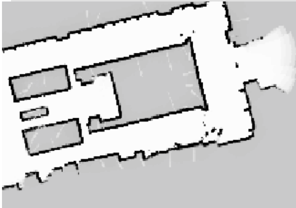
Model Acquisition





courtesy: Wolfram Burgard

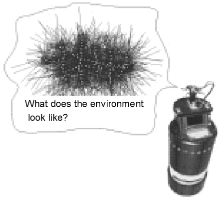
Model Use

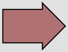


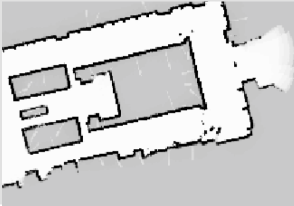
Cognition-enabled Control — the Very Idea

Example: Map Acquisition and Map-based Navigation

Model Acquisition

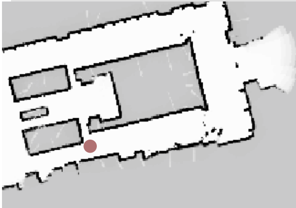






courtesy: Wolfram Burgard

Model Use

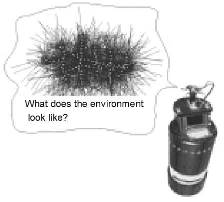


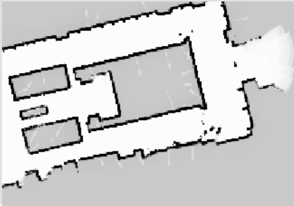
Where am I?

Cognition-enabled Control — the Very Idea

Example: Map Acquisition and Map-based Navigation

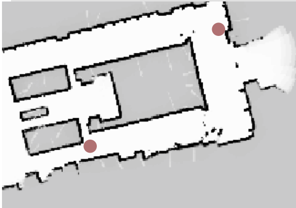
Model Acquisition





courtesy: Wolfram Burgard

Model Use



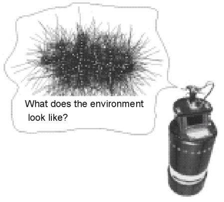
Where am I?

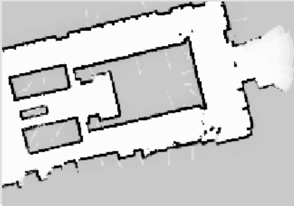
Where is L?

Cognition-enabled Control — the Very Idea

Example: Map Acquisition and Map-based Navigation

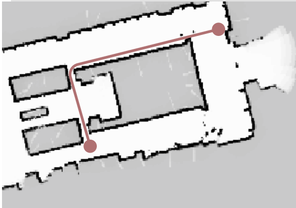
Model Acquisition





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 $\langle tsk \rangle$
  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

Cognitive mechanisms enable us to control the robot

- reliably
- flexibly
- efficiently

in concise control programs

Cognition-enabled Robot Control

A Working Definition

= information processing, perception, and action control 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

Cognition-enabled Robotics in the Housework Domain

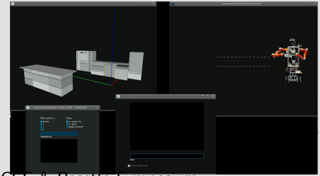
Robotic Roommates Making Pancakes

Our Vision: Robotic workers, co-workers, assistants that can

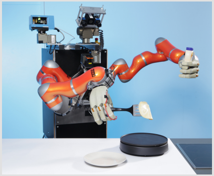
- perform human-scale tasks and jobs;
- execute naturalistic task & action specifications and instructions ;
- perform everyday manipulation;
- extend their repertoire of services by acquiring new skills using information resources intended for human use.

in realistic domestic and factory settings

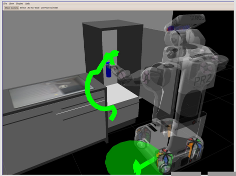
Plan Generation



Plan Execution



Plan Explanation

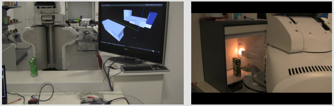


Making “Weisswürste” and Going Shopping

Shopping & cleaning up

1. shopping with basket

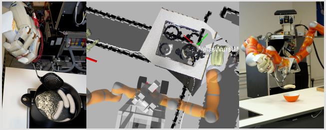
2. clean up according to organizational principles



Making “Weisswürste”

1. putting “Weisswürste” into pot

2. fishing “Weisswürste”



3. cutting bread

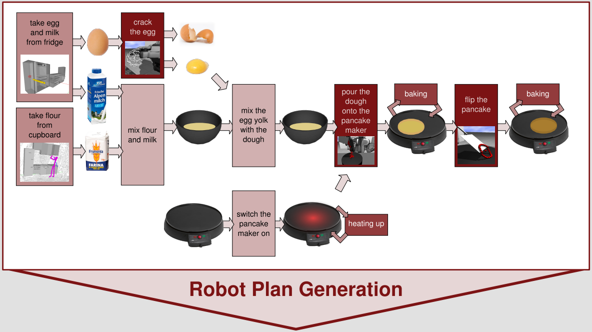
Robots Doing Housework (1)

What are the problems?

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:



Robots Doing Housework (2)

What are the problems?

Naturalistic Action Description

push the spatula under the pancake

Effective Action Specification

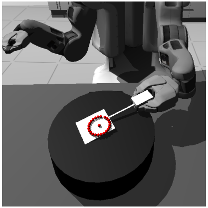
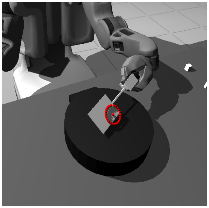
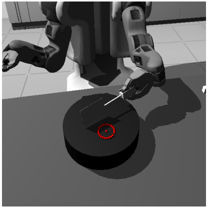
hold the handle of the spatula and push the blade of the spatula under the pancake such that

- you can lift the pancake safely,
- don't damage the pancake, and
- don't push the pancake off the oven

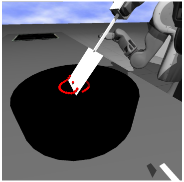
Robots Doing Housework (3)

What are the problems?

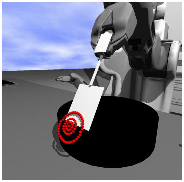
- **Parameters:** angle of spatula
- **Outcomes:** turned, not turned

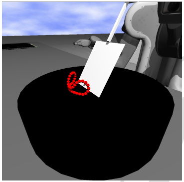
- Common failures:



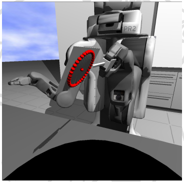
break



push off



fold

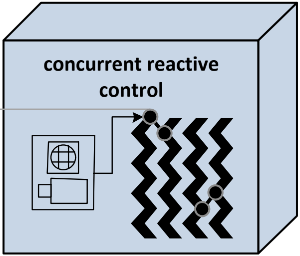


stick on

Principles

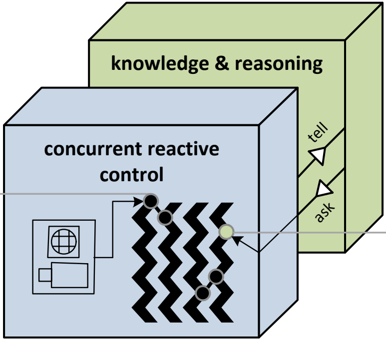
Cognition-enabled Control: Three Principles

Principle 1
Sensor and process events trigger and synchronize concurrent processes.



Cognition-enabled Control: Three Principles

Principle 1
Sensor and process events trigger and synchronize concurrent processes.



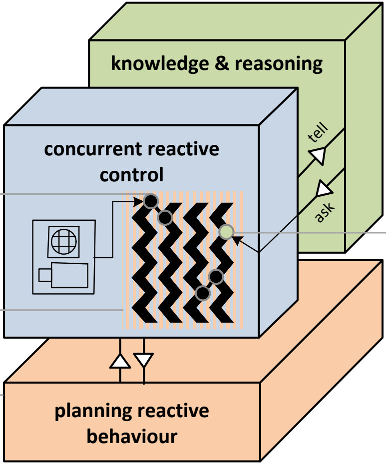
Principle 2
Logical queries infer decisions and parameters using specialized inference methods.

Cognition-enabled Control: Three Principles

Principle 1
Sensor and process events trigger and synchronize concurrent processes.

Principle 3
Large parts of the concurrent control program are plans.

Reasoning and planning mechanisms operate on representations of the concurrent control program and can reason about it and modify it during execution.

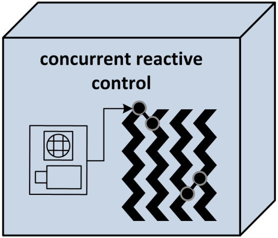


Principle 2
Logical queries infer decisions and parameters using specialized inference methods.

Concurrent Percept-guided Control

Principle 1

Cognition-Enabled Perception-Guided Control Programs



Concurrent, Percept-guided Control

Robot control programs specify how the robot is to respond to percepts and events (failures, etc) to accomplish its goals.

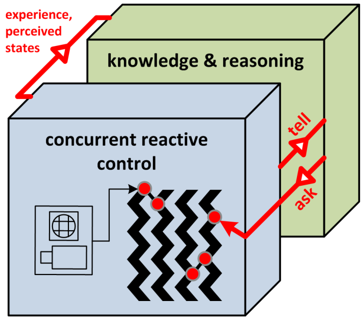
AI Approach	Cognition-enabled Control
plans are (partially ordered) sets of plan steps	plans are concurrent, reactive control programs
actions have preconditions	actions are "universal"
robots have to reason about all the plans	. . . only about plans they generate ensure plans are easy
provably correct plans (optimal, most robust)	improve expected performance
single query property	exploit everyday property

⇝ **Cognition-enabled control can efficiently reason about plans that generate high-performance behavior**

Inference by Plan Statements

Principle 2

Cognition-Enabled Perception-Guided Control Programs



The Need for Specialized Reasoning Methods

Example Programs

- clean up:

  for each object on the table do
  put object where it belongs

- set the table:

  for each object that is needed do
  put object where it belongs

- push the spatula under the pancake:

The Need for Specialized Reasoning Methods

Example Programs

- clean up:
  for each object on the table do
  put object where it belongs
- set the table:
  for each object that is needed do
  put object where it belongs
- push the spatula under the pancake:

Specialized Reasoning

Inference tasks are

- too complex,
- too varied,
- too strongly affected by
  - uncertainty,
  - real-time constraints,
  - real-world conditions

to be addressed by general-purpose reasoning

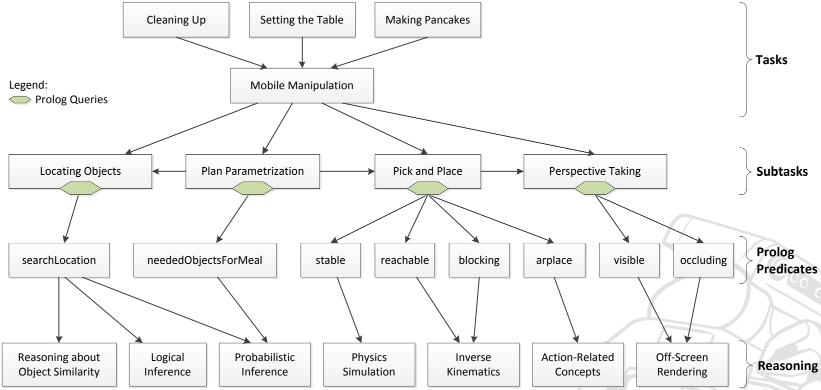
PROLOG as a Uniform Framework

The pose ?pose for putting down an object based on the current and an expected future location of the robot where it will pick it up later.

- **reachable from both of these locations**
- **stable on the kitchen counter**
- **visible from the robot's expected future location.**

objectPose(W, Cup, **?pose***),*	
*on(***?pose***, CounterTop),*	
currentRobotPos(?currPos),	procedural attachment
expectedRobotPos(?expectedPos),	
stable(W, Cup),	physics simulation
reachableFrom(W, ?currPos, Cup),	inverse kinematics
reachableFrom(W, ?expectedPos, Cup),	inverse kinematics
visible(W, ?expectedPos, Cup)	opengl

PROLOG as a Uniform Framework



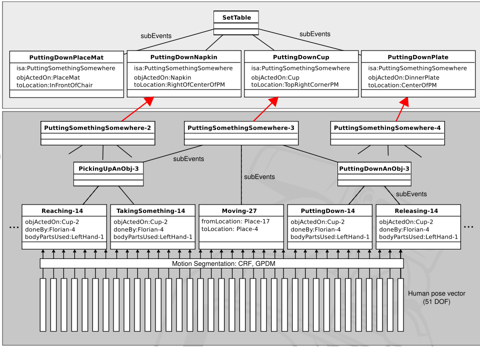
Using the Program as a Knowledge Base

Inference Method 1

Example:
subtask(Super,Sub)

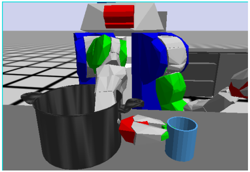
1. subtask(Super, Sub):-
   var(Super), **var**(Sub),
   !, fail.
2. subtask(Super, Sub):-
   var(Super), **nonvar**(Sub),
   Super $\leftarrow$ procCall Sub$\rightarrow$Super(Sub,tasktree)
3. subtask(Super, Sub):-
   nonvar(Super), **var**(Sub),
   Sub $\leftarrow$ procCall SubTask(Super,tasktree)
4. subtask(Super, Sub):-
   nonvar(Super), **nonvar**(Sub),
   Sub $=$ procCall SubTask(Super,tasktree)

Action task tree



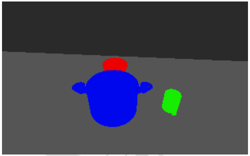
Parameterizing Actions with their Effects

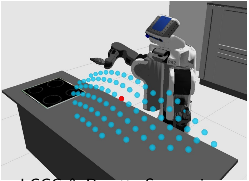
Inference Method 2

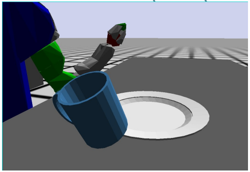


Put the pancake mix away

```
(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))))))))
```







Effect-based Action Parameterization

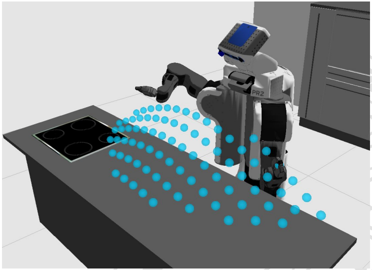
setof ?Pose On(Counter, ?Pose) ?Poses $\wedge$ member(?P, ?Poses)
$\wedge$ Pose(Cup, ?P) $\wedge$ stable(Cup)

Effect-based Action Parameterization

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

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

Create distribution for sampling poses

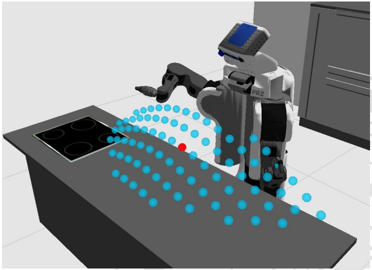


Effect-based Action Parameterization

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

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

Draw a pose sample

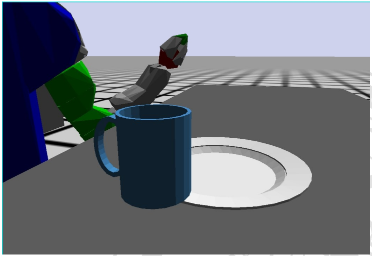


Effect-based Action Parameterization

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

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

Place the mug

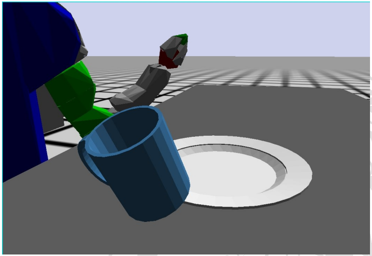


Effect-based Action Parameterization

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

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

Simulate for 50ms, fail!

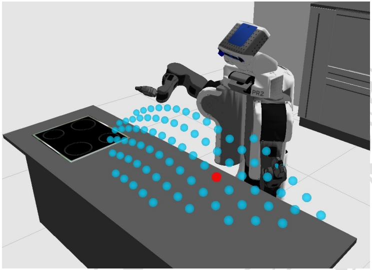


Effect-based Action Parameterization

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

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

Backtrack, draw another pose sample

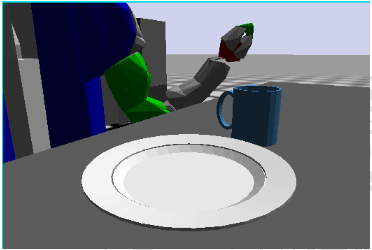


Effect-based Action Parameterization

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

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

Place the mug

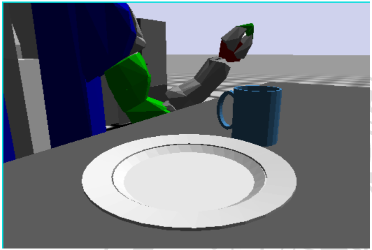


Effect-based Action Parameterization

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

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

Simulate for 50ms, succeed!



Action-related Concepts

Inference Method 3

instead of prespecifying decisions

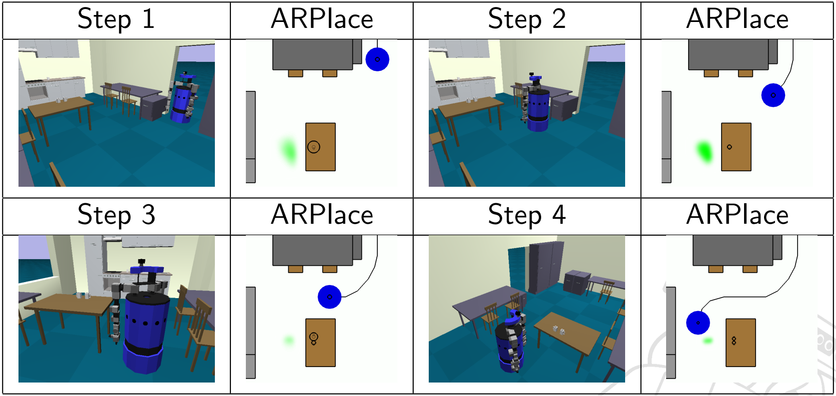
```
(at-location ⟨ OBJ.POS.x - 60, OBJ.POS.y - 10 ⟩
    (pick-up    OBJ))
```

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))
```

Inferring Control Decisions

Lazy, evidence-based decision making

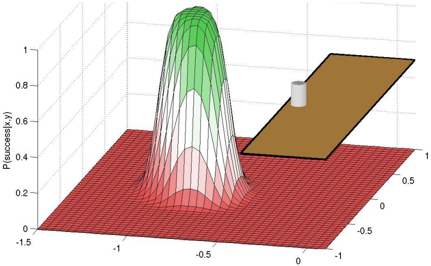


"A decision is a commitment to a plan or an action parameterization based on evidence and the expected costs and benefits associated with the outcome."

adapted from Resulaj et al, *Changes of mind in decision-making*

Learning Action-related Places

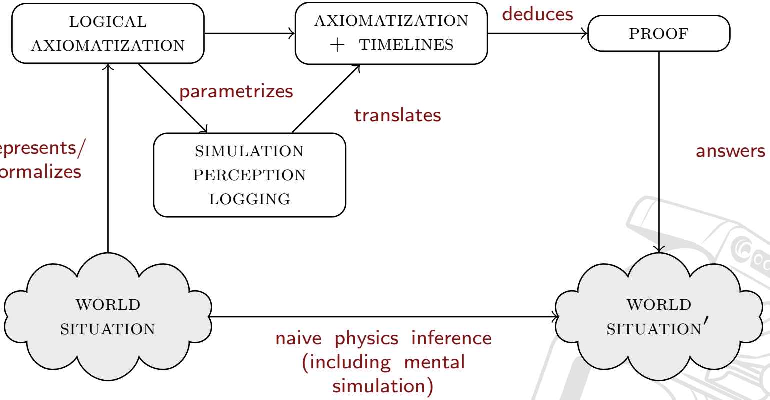
- Representation:
  - Discretized space of potential maniplation places
  - Mapping to expected utilities



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

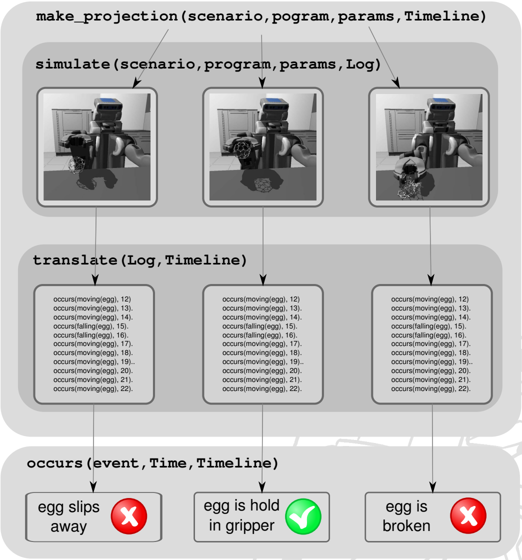
Simulation-based Temporal Projection

Inference Method 4



Temporal Projection Process

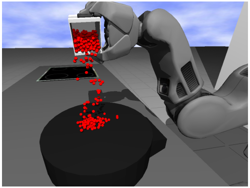
- **make_projection:**
  sample parameters
  - **simulate:**
    setup simulator
    run simulation
  - **translate:**
    ground predicates
    in logged
    simulations
- **evaluate:**
  events/fluents
  specialized predicates



Example: Making a Pancake

Pouring

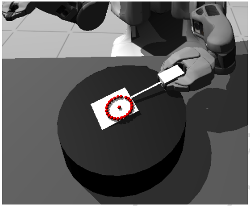
- Parameters: position, time, angle
- Outcomes: number of particles on pan (spilled on table)



- Specialized predicates on particle sets: round/centered

Flipping

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



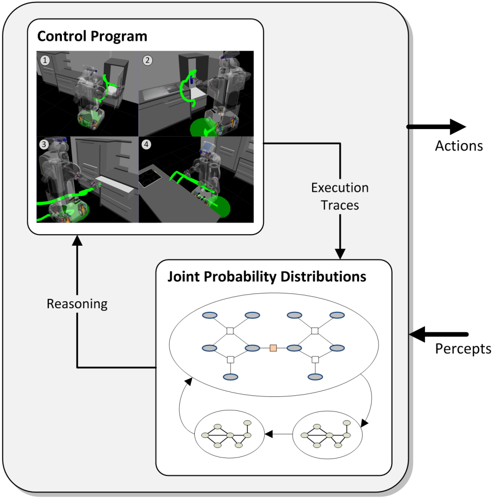
Common failures:

- break, push off, fold, stick on

⇝ **Parameters that lead to desired outcomes are inferred**

Bayesian Cognitive Robotics

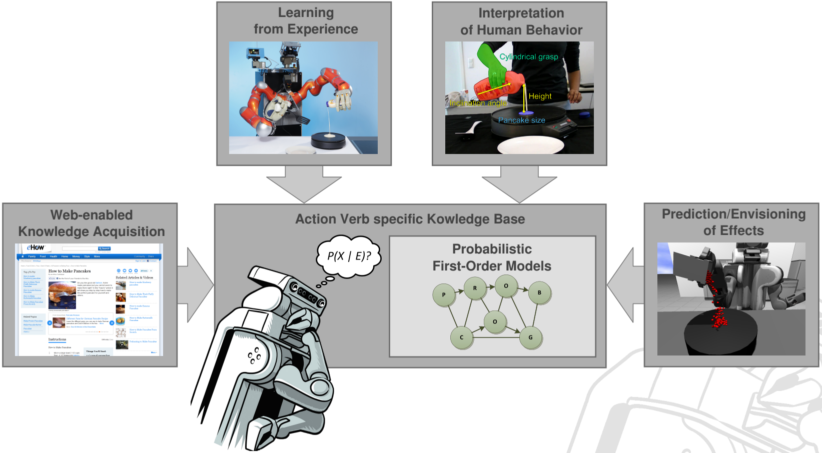
Inference Method 5



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

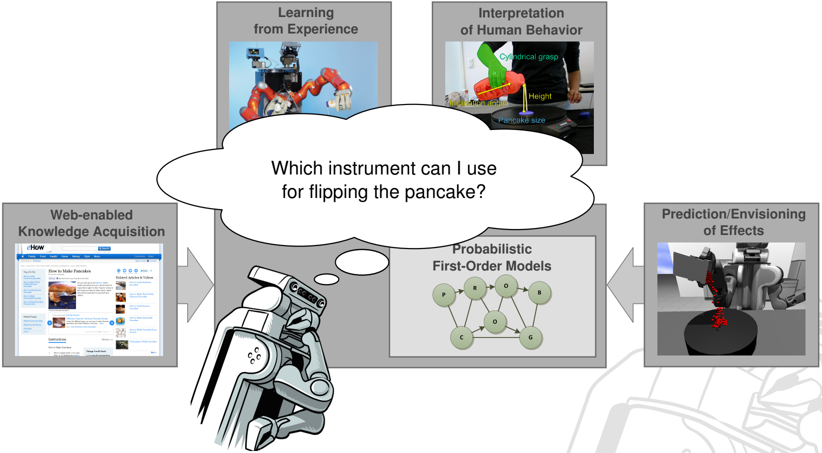
An Action Verb specific Knowledge Base

Sources of Knowledge and Cognitive Capabilities



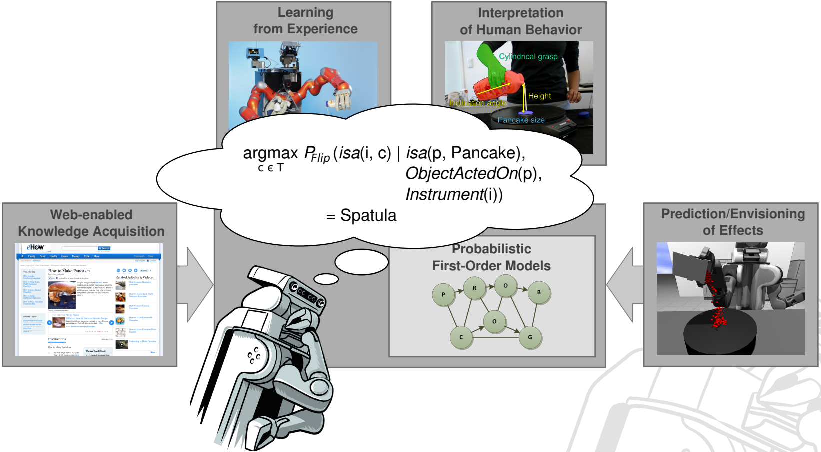
Example

"Flip the pancake!"



Example

"Flip the pancake!"



$$\underset{c \in T}{\operatorname{argmax}} P_{Flip}(isa(i, c) \mid isa(p, \text{Pancake}), ObjectActedOn(p), Instrument(i)) = \text{Spatula}$$

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**

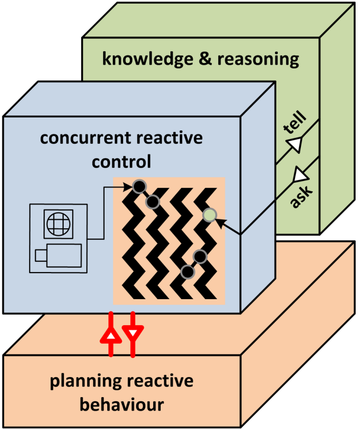
  P(localizationQuality(Robot, Bad, Sit) |
  ¬successful(Robot, Grasp, Obj, Sit) ∧ . . .)

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

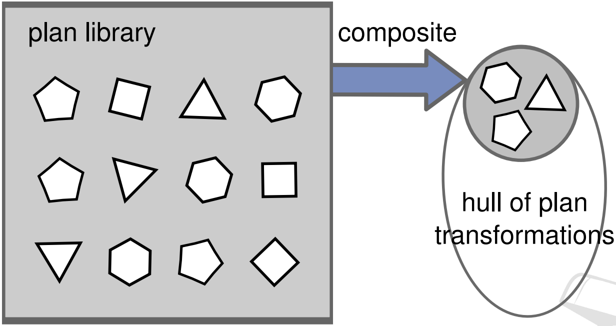
Plan-based Robot Control

Principle IV

Cognition-Enabled Perception-Guided Action Plans



Operational Definition of the Plan Language

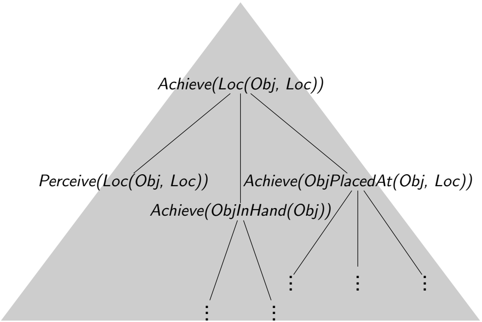


All plans have property **p** if

- the plan schematas in the plan library satisfy p
- plan composition preserves p
- plan transformation preserves p

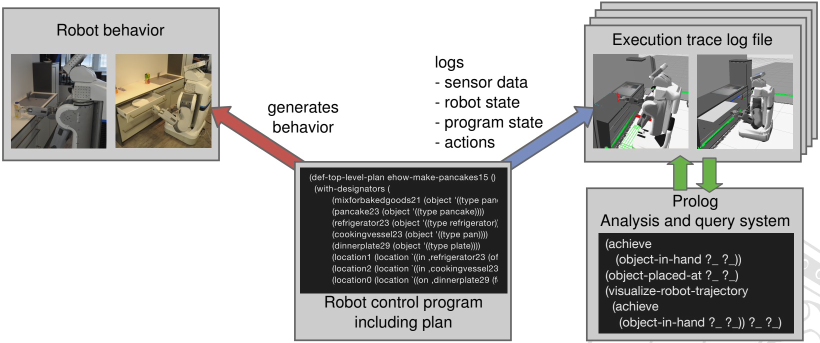
Transparent Plan Property

Declarative Goal Hierarchies



- the plan is structured into code pieces that have the names achieve(g), perceive(p), maintain(g), ...
- if a plan segment is named achieve(g) if and only if it is intended to achieve g

Reasoning about Execution Traces



Conclusions

- **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

Selected Next Steps

- movement as first-class objects (symbolic: constraints and objective functions, subsymbolic: iTASC, Stack of Tasks
- imitation learning through physics-based interactive games
- learning action-based knowledge bases (from the web, from experience)
- performing generalized pick and place for a week
- Bayesian Cognitive Robotics
- web-enabled Robots
- robot crowd sourcing
- imitation learning with deep task and physics models

Thank you for your attention

Thanks to:

roboh w

SAPHARI

SAFE AND AUTONOMOUS PHYSICAL HUMAN-AWARE ROBOT INTERACTION

RoboEarth

TUM ROS Package Repository:

http://www.ros.org/wiki/tum-ros-pkg

Contact:

http://ias.cs.tum.edu/