



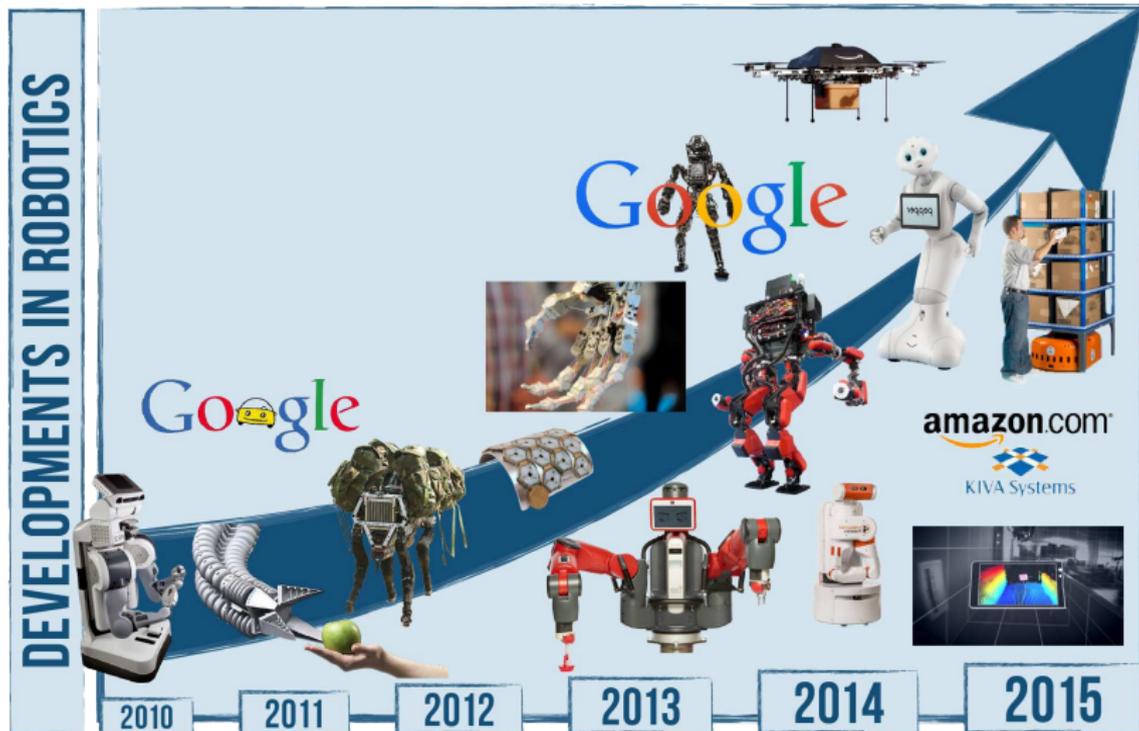
openEASE —
A digital innovation platform for intelligent robotics

Michael Beetz,
University of Bremen

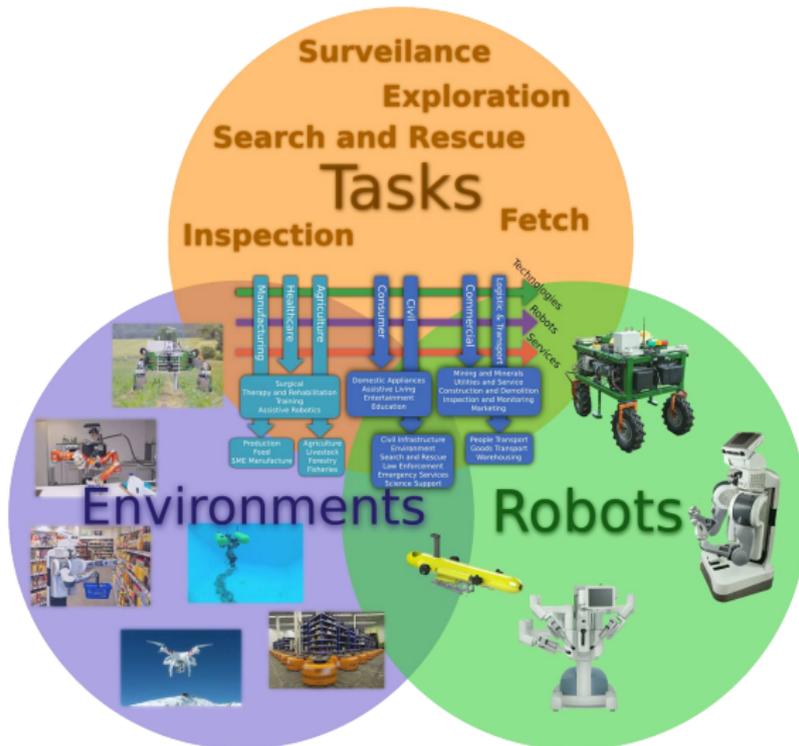
ROS Industrial Conference, November 2016



Robotics Curve



Market requires many robots in many environments performing many tasks



The role of knowledge for robotics

Ginni Rometty (IBM):

- **“Data is the world’s great new natural resource. What steam power was to the 18th century, electromagnetism to the 19th and fossil fuels to the 20th; data will be to the 21st.”**

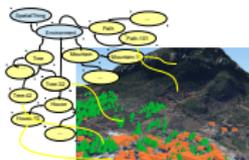
Gill Pratt (Toyota Research Institute, *Is a Cambrian Explosion Coming for Robotics?*. Journal of Economic Perspectives, Vol. 29, No. 3 (Summer 2015)):

- **“Robots are already making large strides in their abilities, but as the generalizable knowledge representation problem is addressed, the growth of robot capabilities will begin in earnest, and it will likely be explosive.”**
- **“The key problems in robot capability yet to be solved are those of generalizable knowledge representation and of cognition based on that representation.”**

Why Is Knowledge so Important?

the cost of not knowing

- if the robot does not know about the task, the environment, the robot the programmer has to hardcode everything



- programming/instructing at an abstract/semantic level
 - put the bolt into the nut and fasten it
 - pour water into the glass
 - ...

Wikipedia:
Screw



Challenge 1:

Closing knowledge gaps

Action description

pour the water out of the pot
(perform
 (an action
 (type pour)
 (theme water)
 (source pot)))



infer motion parameters and constraints such as

- 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

Challenge 2:

Symbolic action descriptions cause different behavior

pouring plan

```
def-plan pour ( (theme) : (some stuff)
               (source) : (an object
                           (type container)
                           (contains (theme))
                           (affordance (an action
                                         (type pick-up)
                                         (body-part (a bo
                                                    (type hand))))))
               (dest) : (a location))

begin
  1. reach( (source) ) /* (includes grasp, pregrasp)
  2. lift( (source) ) (a location (above (destination)))
  3. tilt( (source) )
  until (amount (some stuff (at (destination))))
      ≥ (amount (theme))
end
```



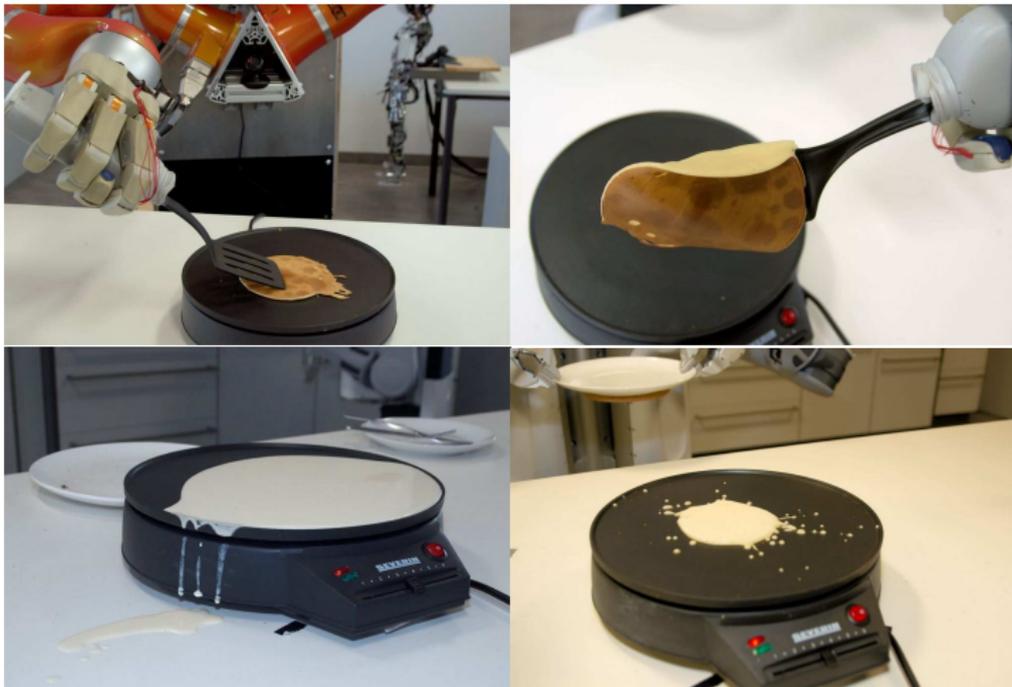
action description

(perform
 (an action
 (type adding)
 (theme (some substance
 (type milk))))
 (destination
 (some dough))))))



Challenge 3:

Action success requires motion skills



A generalized action plan for pouring

```
def-plan pour ( <theme> : (some stuff)
                <source> : (an object
                            (type container)
                            (contains <theme>))
                            (affordance (an action
                                        (type pick-up)
                                        (body-part (a body-part (type hand))))))
                <dest> : (a location))
```

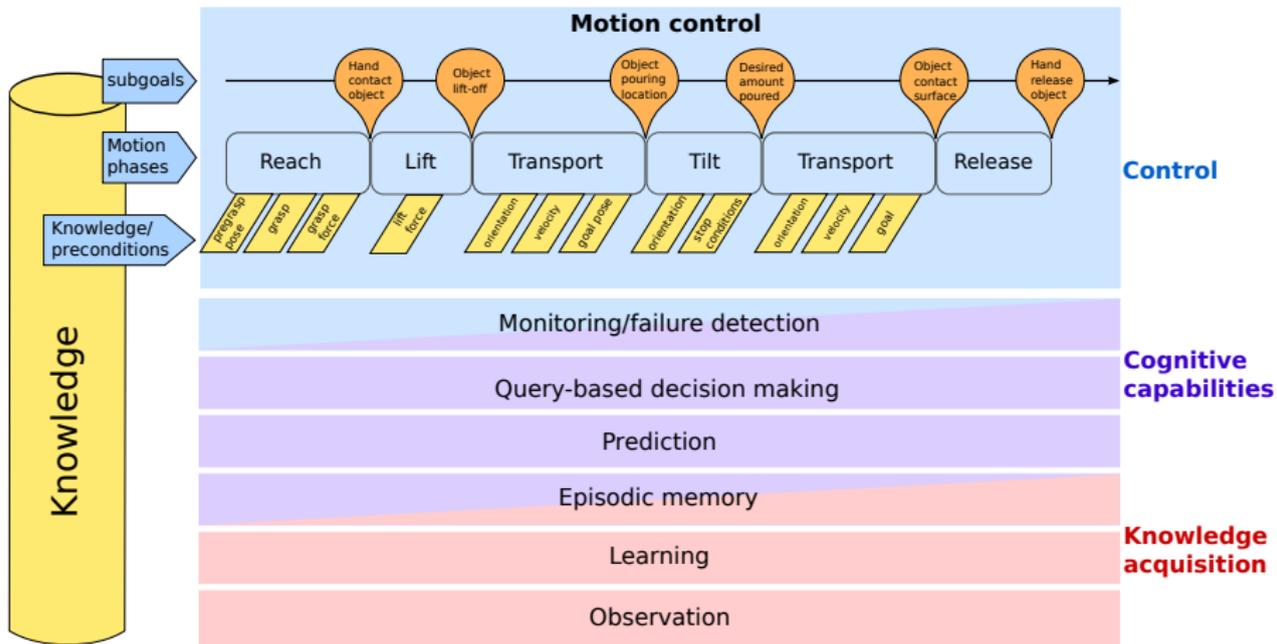
begin

1. reach(<source>) /* (includes grasp, pregrasp)
2. lift(<source>) (a location (above <destination>))
3. tilt(<source>)
until (amount (some stuff (at <destination>)))
≥ (amount <theme>))

end

Generalized Cognition-enabled Motor Plan/Program

inspired by [Flanagan]



ROBOPAL — a personal assistant for robots



- **given:** I want to pour water into the cup
- **question:** how should I grasp and hold the cup

This is not science fiction!!!

look at Google Home, Amazon Echo, Siri, Viv, ...



ROBOPAL — a closer look

how should I grasp the cup?

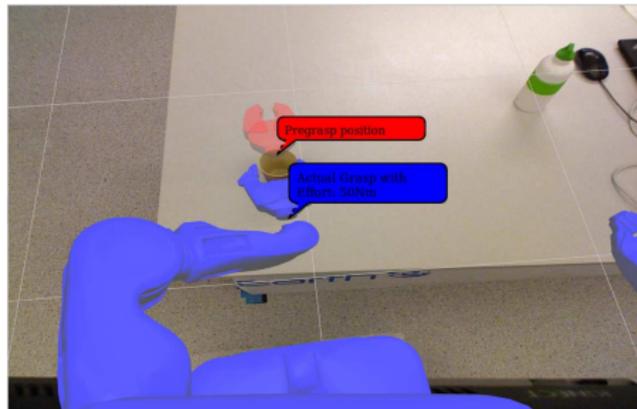
```
?- current-task(Tsk),
   task-exp(Tsk,
     [an, action,
      [type, pouring],
      [destination, [an, object,
                    [type, container]]],
      [support-action, Spt-Tsk]]),
   task-exp(Spt-Tsk,
     [an, action,
      [type, grasping],
      [subactions, Motions]]),
   subaction([a, motion
             [type, reaching],
             [pregrasp-pose, PG-pose],
             [grasp-pose, G-pose],
             [grasp-force, G-force]],
             Tsk),
   show(PG-pose,G-pose,G-force).
```



ROBOPAL — a closer look

how should I grasp the cup?

```
?- current-task(Tsk),
   task-exp(Tsk,
     [an, action,
      [type, pouring],
      [destination, [an, object,
                    [type, container]]],
      [support-action, Spt-Tsk]]),
   task-exp(Spt-Tsk,
     [an, action,
      [type, grasping],
      [subactions, Motions]]),
   subaction([a, motion
              [type, reaching],
              [pregrasp-pose, PG-pose],
              [grasp-pose, G-pose],
              [grasp-force, G-force]],
              Tsk),
   show(PG-pose, G-pose, G-force).
```



Knowledge needed for answering

rule base

- if a container is filled & open then hold it upright
- if an object can break then don't squeeze too hard
- if the task context is pouring then grasp close to the com
- if the task context is pouring then don't grasp mouth

rule base (ctd)

- **choose motion parameters that are predicted to succeed**
- grasp an object such that you have good visual feedback
- don't get too close to breakable objects

Knowledge needed for answering

many motion constraints are semantic
many apply in many situations
relevant ones might be inconsistent

rule base

- if a container is filled & open then hold it upright
- if an object can break then don't squeeze too hard
- if the task context is pouring then grasp close to the com
- if the task context is pouring then don't grasp mouth

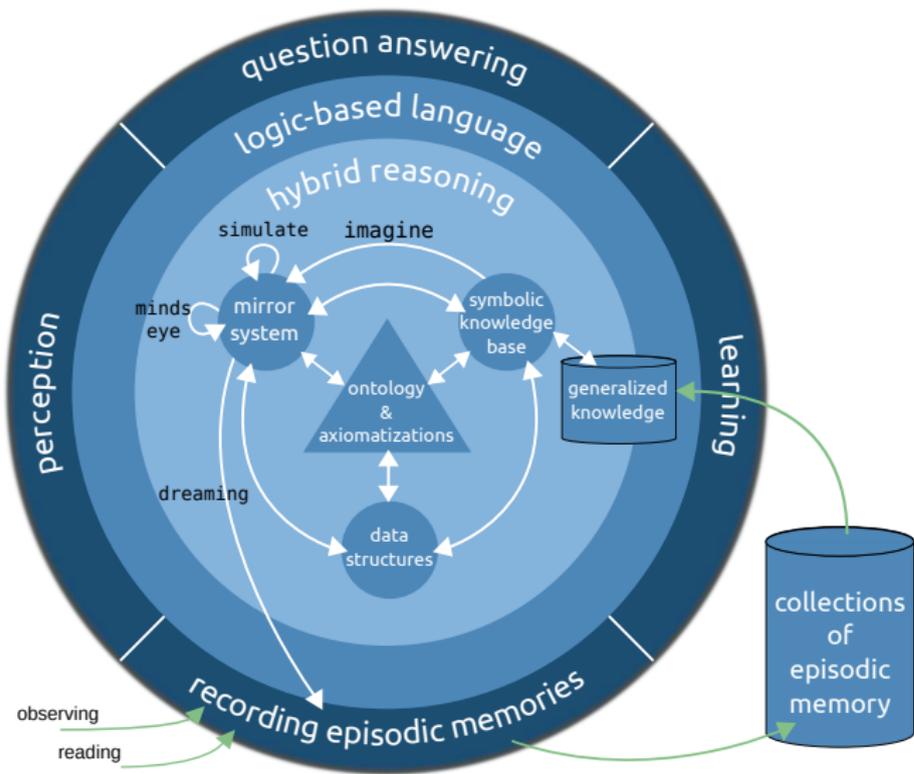
rule base (ctd)

- choose motion parameters that are predicted to succeed
- grasp an object such that you have good visual feedback
- don't get too close to breakable objects

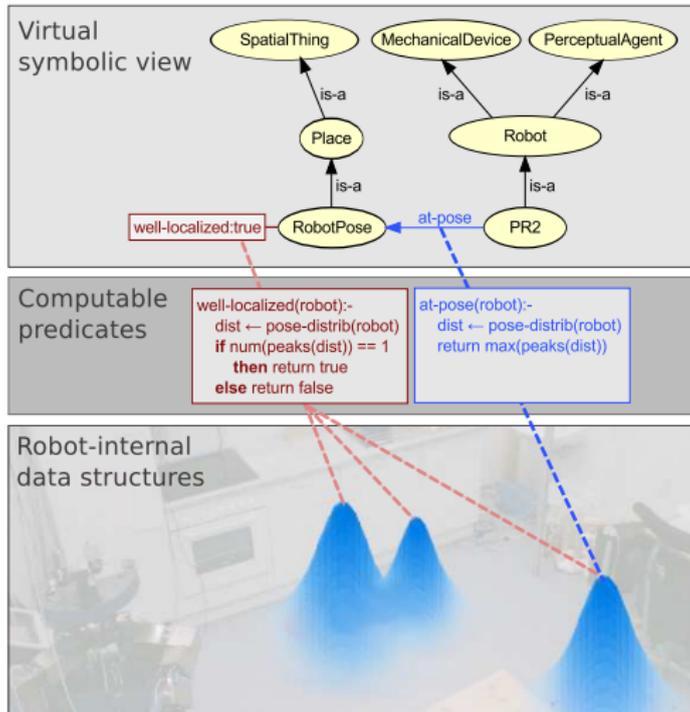
Knowledge-enabled programming

fetch-and-place plan schema	knowledge
<pre><u>def-plan</u> pour (⟨theme⟩, ⟨source⟩ ⟨destination⟩) 1. take(⟨source⟩) 2. hold(⟨source⟩) (a location (above ⟨destination⟩)) 3. tilt(⟨source⟩) <u>until</u> (amount (some stuff (at ⟨destination⟩))) ≥ (amount ⟨theme⟩) <u>end</u></pre>	<ul style="list-style-type: none">● filled open containers must be held upright● soft objects must not be squeezed beyond their limits● an open tetrapak is soft● you have to tilt a pot around the axis between the handles● when tilting a container with an unconnected lid then the lid might fall down● objects with handles are conveniently held by their handles● objects that are to be picked up with two hands must be reachable by both hands● ...

Knowledge processing for robots



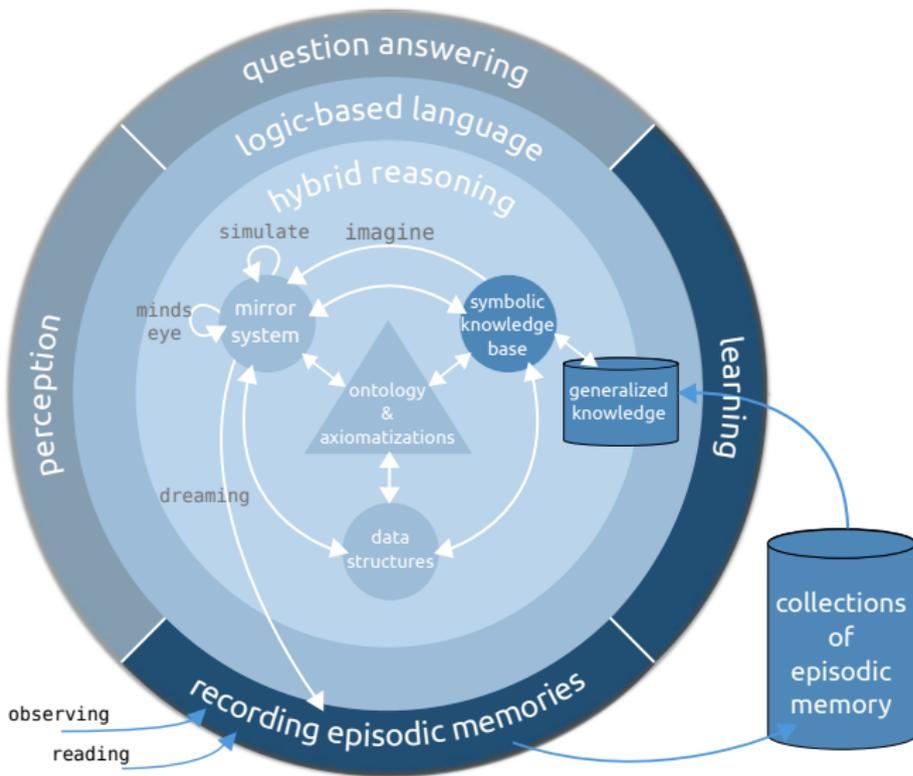
Virtual knowledge bases



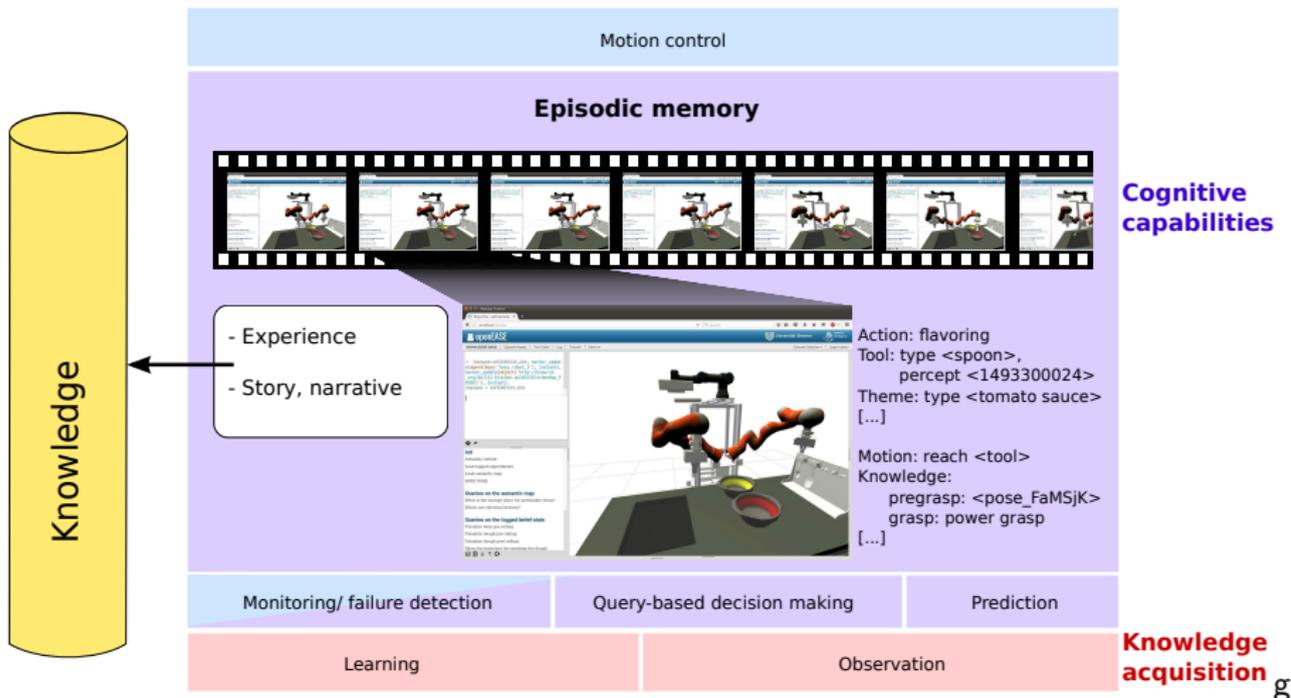
Mirror world knowledge bases



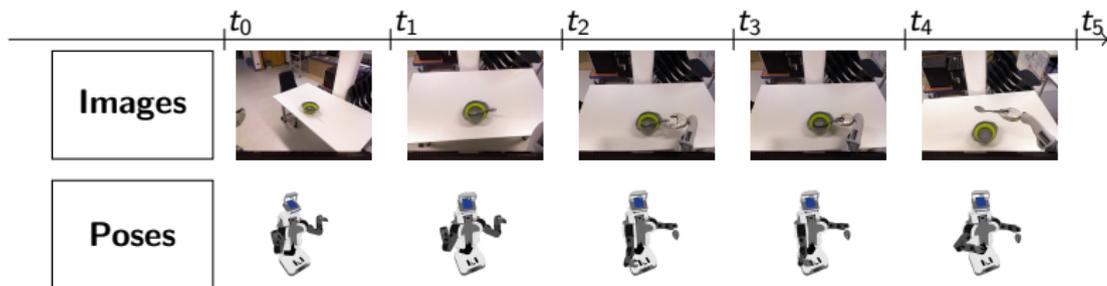
Knowledge acquisition and episodic memories



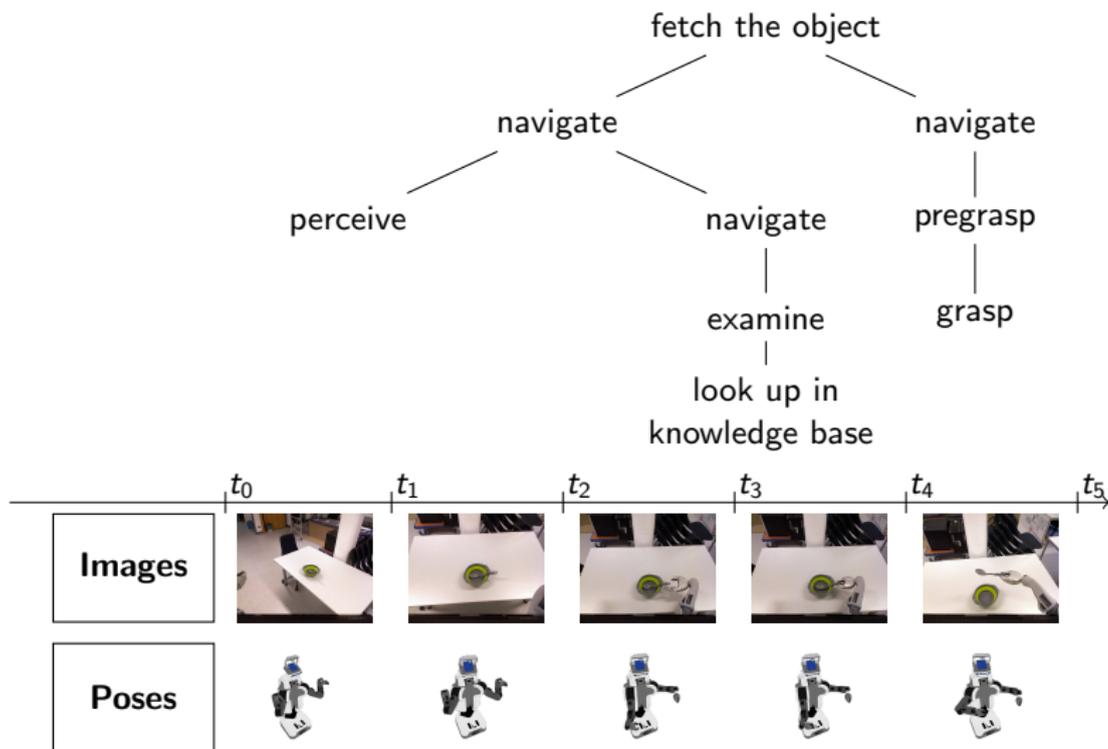
Episodic memories



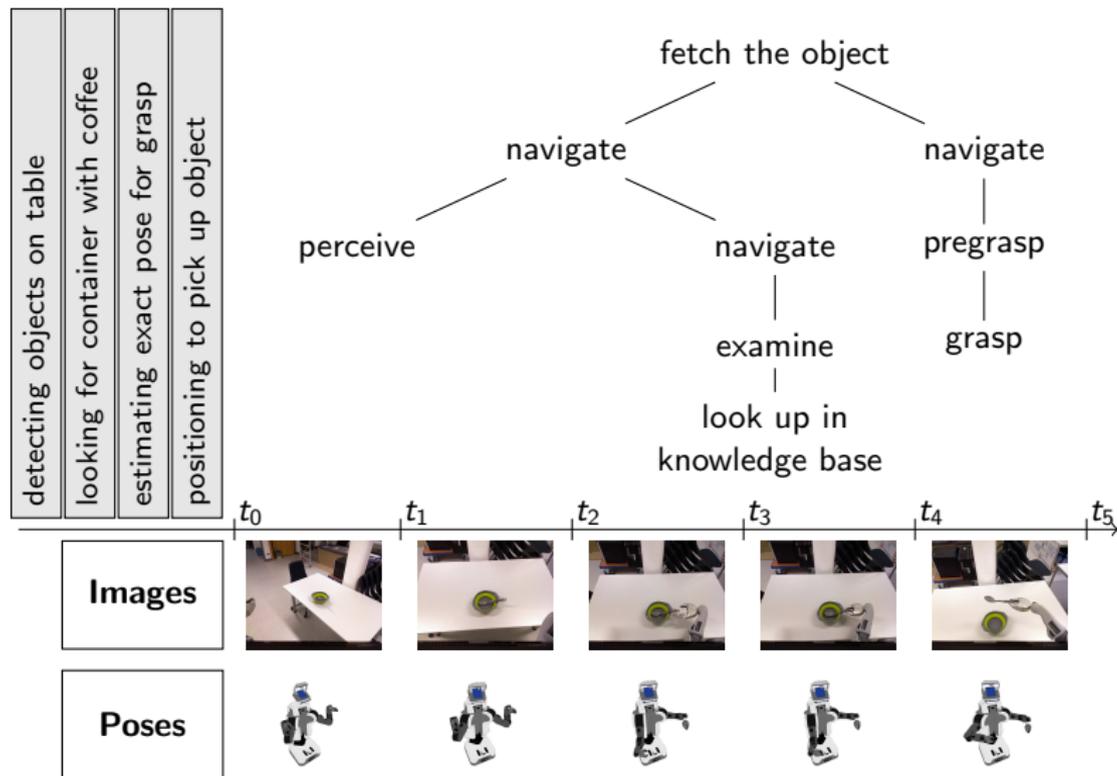
Components of episodic memories



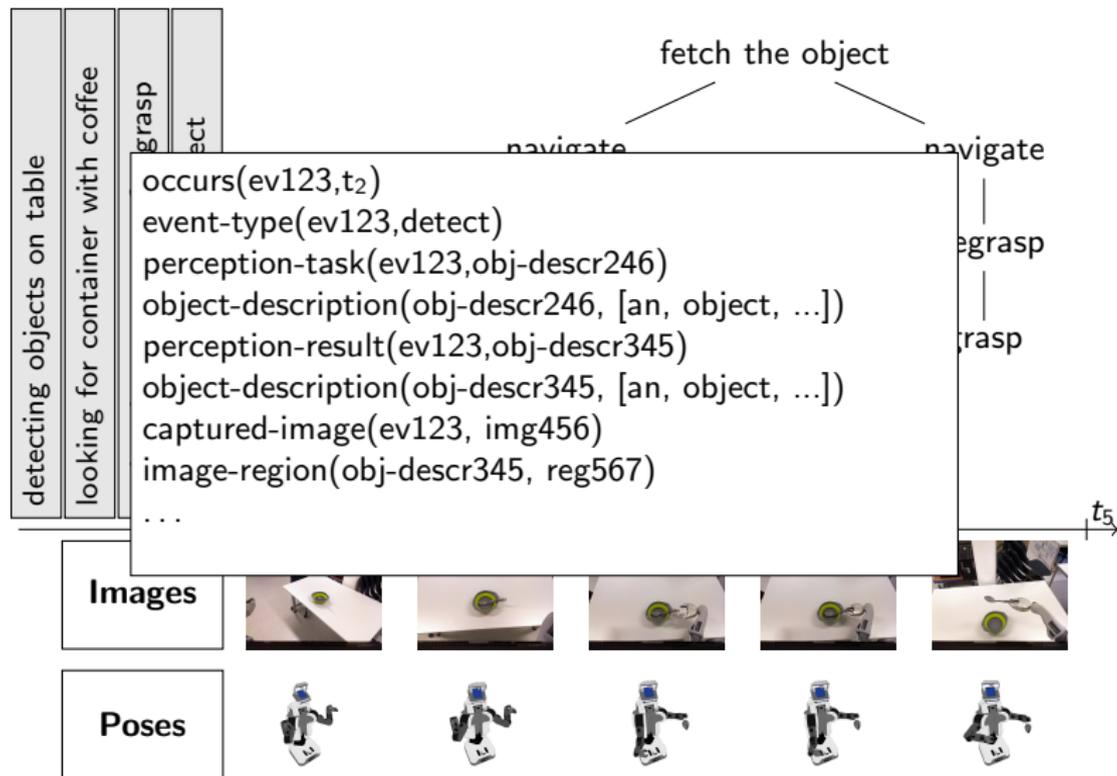
Components of episodic memories



Components of episodic memories



Components of episodic memories



Semantic retrieval from episodic memories

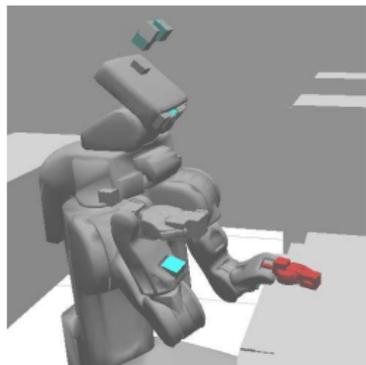
```
?- task(Tsk),  
   task-action(Tsk, [an, action,  
                   [type, pick-up],  
                   [object-acted-on,  
                    [an, object  
                    [type, pot],  
                    [weight, Weight]]]]),  
   Weight >= 2kg,
```

Semantic retrieval from episodic memories

```
?- task(Tsk),
   task-action(Tsk, [an, action,
                    [type, pick-up],
                    [object-acted-on,
                     [an, object
                      [type, pot],
                      [weight, Weight]]]]),
   Weight >= 2kg,
task-start(Tsk, TskStrt),
```

Semantic retrieval from episodic memories

```
?- task(Tsk),  
   task-action(Tsk, [an, action,  
                    [type, pick-up],  
                    [object-acted-on,  
                     [an, object  
                      [type, pot],  
                      [weight, Weight]]]]),  
   Weight >= 2kg,  
   task-start(Tsk, TskStrt),  
   holds(pose(pr2, Pose), TskStrt).
```



Supervised learning from episodic memories

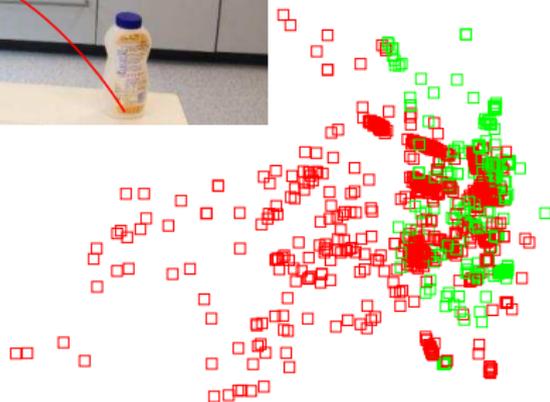
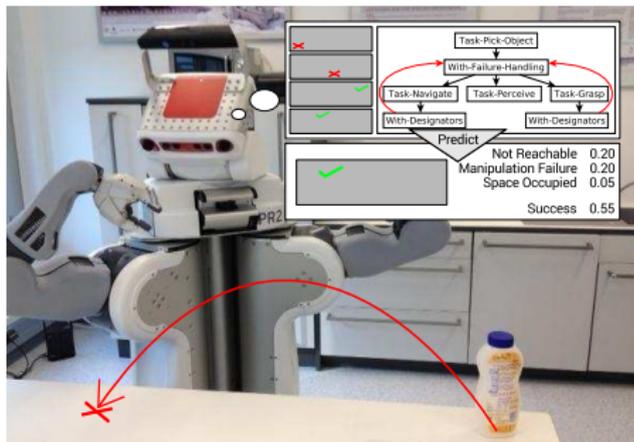
?-

```
task(Tsk),  
task-action(Tsk, [an, action,  
                 [type, pick-up],  
                 [object-acted-on,  
                 [an, object  
                 [type, pot],  
                 [weight, Weight]]]]]),  
  
Weight >= 2kg,  
  
task-start(Tsk, TskStrt),  
holds(pose(pr2, Pose), TskStrt),  
.
```

Supervised learning from episodic memories

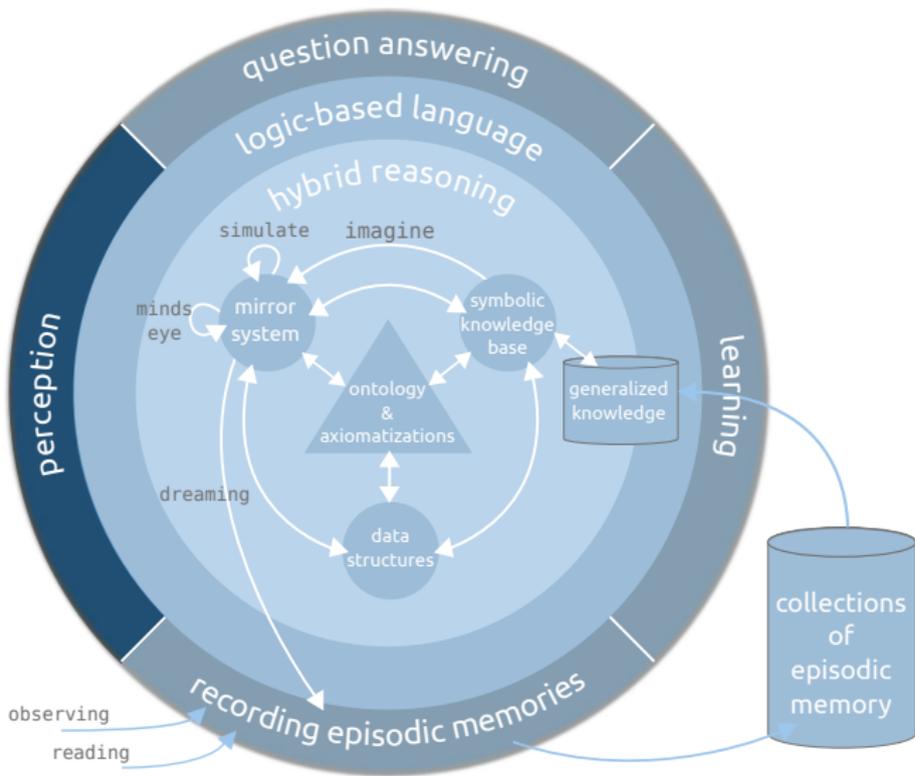
```
?- setof(Pose,  
        task(Tsk),  
        task-action(Tsk, [an, action,  
                        [type, pick-up],  
                        [object-acted-on,  
                        [an, object  
                        [type, pot],  
                        [weight, Weight]]]]]),  
        Weight >= 2kg,  
        task-outcome(Tsk, success),  
        task-start(Tsk, TskStrt),  
        holds(pose(pr2, Pose), TskStrt),  
        Poses).
```

Learning control concepts from longterm experience

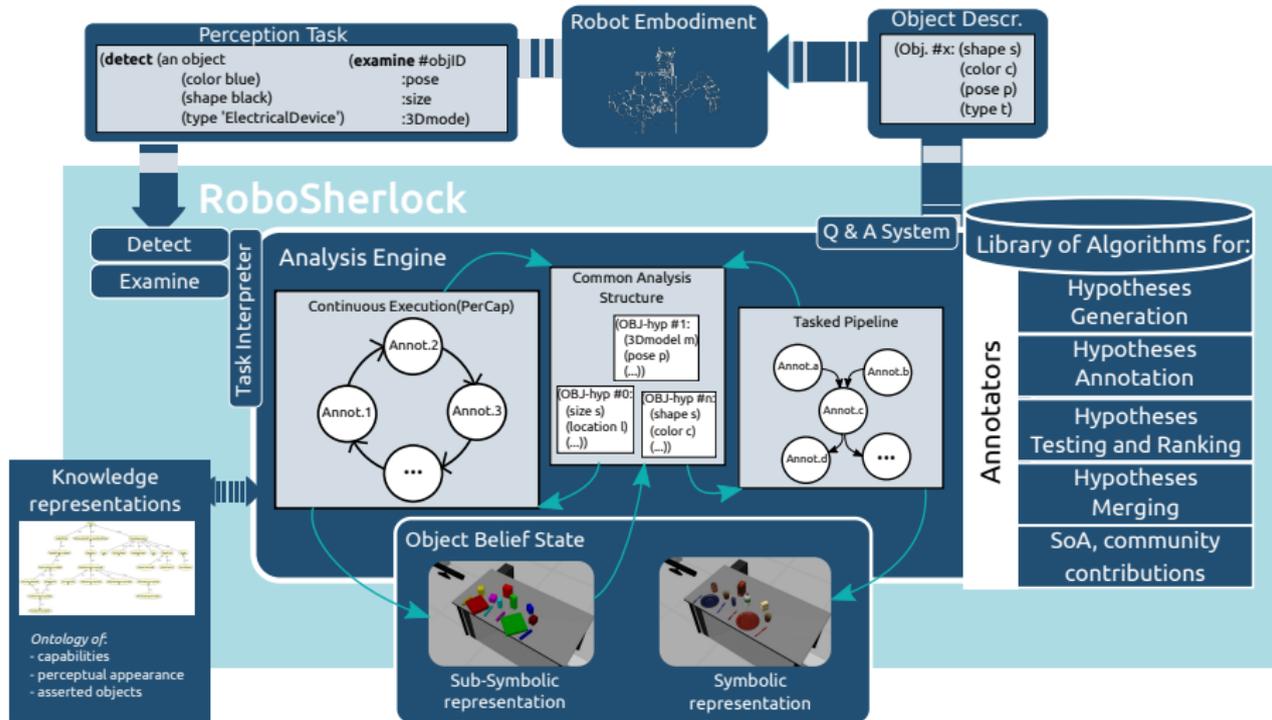


Counter Top

Perception



Cognition-enabled perception



Final remarks

`openease/scenario-components.png`

Final remarks (2)



Open Knowledge for AI-Enabled Robots

open-ease.org

The screenshot shows the openEASE website with several callouts highlighting key features:

- Knowledge Base Graphical User Interface**: A callout pointing to a 3D visualization of a robotic arm in a virtual environment.
- tutorials**, **installation guide**, and **documentation**: A callout pointing to the navigation menu.
- open access to all users**: A callout pointing to the 'Register' button in the navigation menu.
- Knowledge Base Editor**: A callout pointing to a code editor interface showing XML-like code.
- Acquiring everyday manipulations**: A callout pointing to a video thumbnail showing a person interacting with a robot.
- open teaching**: A callout pointing to a diagram of a robotic arm with joints labeled q_1, q_2, q_3 and L_1, L_2, L_3 .

OPENEASE: other tools

open research/innovation

The screenshot shows the OpenEASE website. At the top, there are logos for OPENEASE, ROS Industrial, and the University of Bremen. The main heading is "Open Knowledge for AI Enabled Robots". Below this, there are sections for "About openEASE" and "Knowledge Base Experiments". The "About openEASE" section describes the project's goal of creating a common knowledge base for AI-enabled robots. The "Knowledge Base Experiments" section features three images: a robot arm, a robot head, and a robot torso, each with a corresponding label: "Integrating ROS into ROS2 (ROS2)", "Robot Head for ROS2", and "Preparation for example knowledge".

probabilistic reasoning

The screenshot shows the pracmin website. At the top, there are logos for pracmin and the University of Bremen. The main heading is "Markov Logic Networks in Python". Below this, there is a section for "About" which describes the tool as a Python library for Markov Logic Networks. The "About" section also includes a list of "Dependencies" and "Features".

knowledge processing

The screenshot shows the KnowRob website. At the top, there are logos for KnowRob, ROS Industrial, and the University of Bremen. The main heading is "KnowRob". Below this, there are sections for "About", "Big Installation", "Documentation", "API", "Support", "Ontologies", and "Publications". The "About" section describes the project's goal of creating a common knowledge base for AI-enabled robots. The "Big Installation" section provides instructions for installing the tool. The "Documentation" section provides links to the documentation. The "API" section provides links to the API documentation. The "Support" section provides links to the support page. The "Ontologies" section provides links to the ontologies. The "Publications" section provides links to the publications.

perception

The screenshot shows the RoboSherlock website. At the top, there are logos for RoboSherlock and the University of Bremen. The main heading is "RoboSherlock". Below this, there is a section for "Welcome To RoboSherlock" which includes a "Table of Contents" with links to "Overview", "Installation and Setup", "Tutorials", "Publications", and "About". The "Table of Contents" section also includes a list of "Publications" and "About".

Michael Beetz

ROS Industrial, November 2016

OPENEASE

Thank you

for your attention

Pepper at AI/Uni-Bremen

First integration during the final RoboHow review (EU project).



Pepper at AI/Uni-Bremen (2)



Pepper at AI/Uni-Bremen (3)

- Integration into the kitchen environment
- First attempts at localization using infrared markers (Optitrack tracking system), but would like to replace it by on-board localization when available.
- Communication from the ROS-World using a bridge, to explain OpenEASE data to participants