Cognition and Interactive Systems

Shiwali Mohan

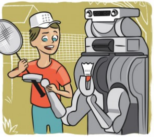
Soar Group
Computer Science and Engineering
University of Michigan

April 10th, 2014

Autonomous Embodied Agents

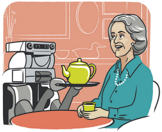
Intelligent behavior in novel environments

recognize and categorize objects

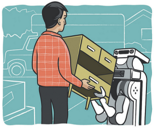


use complex skills

understand and use spatial relationships



perform complex tasks



collaborate with humans

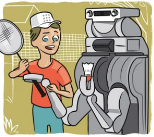
- Diversity in tasks, environments, user preferences
- All usage cannot be predicted at design time
- Designers cannot pre-program all knowledge, skills
- Agents must learn online, continuously, robustly

[1] images from www.willowgarage.com

Autonomous Embodied Agents

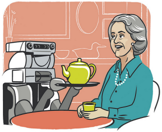
Intelligent behavior in novel environments

recognize and categorize objects

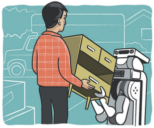


use complex skills

understand and use spatial relationships



perform complex tasks

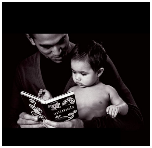


collaborate with humans

- Diversity in tasks, environments, user preferences
- All usage cannot be predicted at design time
- Designers cannot pre-program all knowledge, skills
- Agents must learn online, continuously, robustly
- How to design *taskable* agents?

[1] images from www.willowgarage.com

How do Humans Learn?









Why is Interaction Useful for Learning?

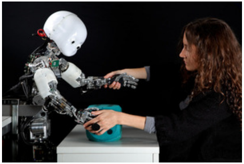
Why is Interaction Useful for Learning?

- Our world is very complex.
- Learning by exploration is slow (and painful, more so if you are an RL agent)
- Interaction
  - reduces perceptual complexity by guiding attention to relevant elements
  - reduces semantic complexity by identifying relevant features
  - encourages discovery of causality between actions
  - helps in knowledge integration
  - validates new knowledge
  - communicates desired/undesired states
  - facilitates incremental learning by scaffolding

Interactive Learning for Agents

Learning from Demonstration
(LfD)

[B. Argall, Northwestern U]



kinesthetic training
teleoperation

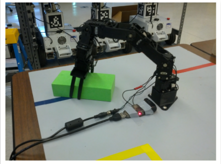
Interactive Reinforcement Learning
(iRL)

[A. Thomaz, Georgia Tech]

reward
feature selection

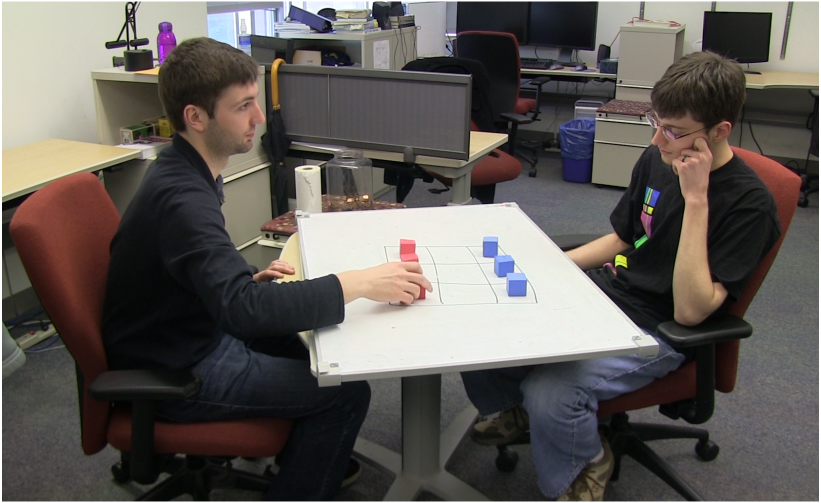
Situated Interactive Instruction
(SII)

[The Soar Group, UM]



natural language
dialog, speech

Interactively Learning a Game



Analyzing Situated Interactive Instruction

Situated language

Instructor: *Its played on a 3-x-3 board.*

Analogical reference

Instructor: *The pieces of the game are basically pawns from chess.*

Mixed-initiative

Instructor demonstrates the game actions to identify the exclusion criterion from the moves available for chess.
Student contributes by the identifying the inclusion criterion.

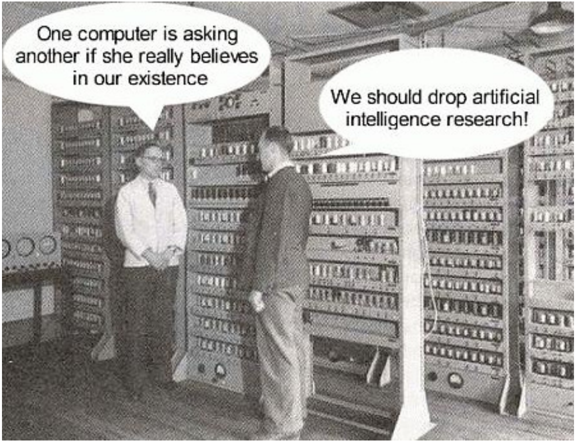
Demonstration

Instructor sets up board, demonstrates initial state/goal state, actions etc.

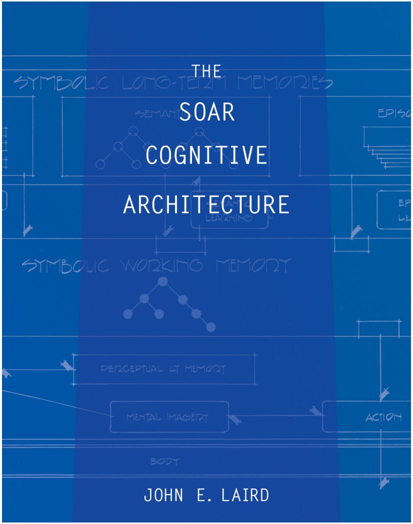
Designing Agents for SII

- Integrated capabilities
  - *Interaction*: speech recognition/generation, grounded language processing, dialog management, non-verbal communication models, eye-tracking and gaze monitoring ...
  - *Traditional AI*: decision making, planning, inductive concept acquisition, deductive problem solving, learning, memory, knowledge representation and reasoning ...
  - *Robotics*: perception in noisy environment, action control and execution ...
- Behavioral properties
  - general, autonomous, knowledge-rich, adaptive, long-living, real-time, robust

AI complete!


One computer is asking another if she really believes in our existence
We should drop artificial intelligence research!

Cognitive Architectures



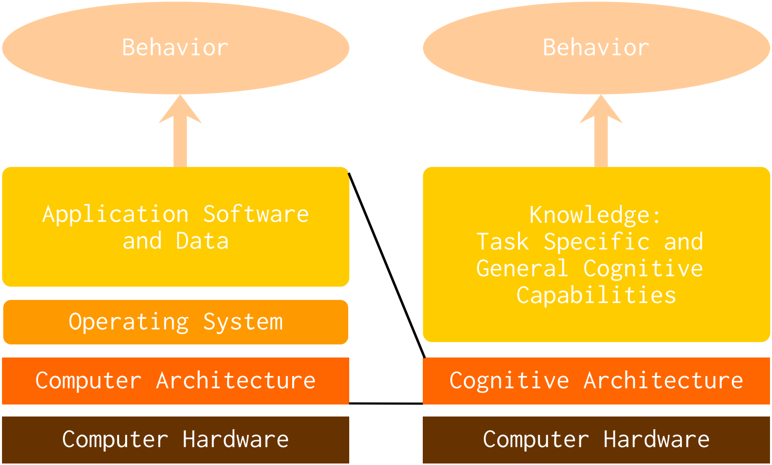
Cognitive Architecture

Goal: Develop and understand human-level intelligence across a diverse set of tasks and domains

Approach: Specify aspects of cognition that are constant for agent lifetime (*cognitive infrastructure*)

- interfaces to perception and action
- memory systems for agent's beliefs, goals, experience
- representation of information and knowledge
- decision making and learning

Computer and Cognitive Architectures



Different Goals of Cognitive Architecture Research

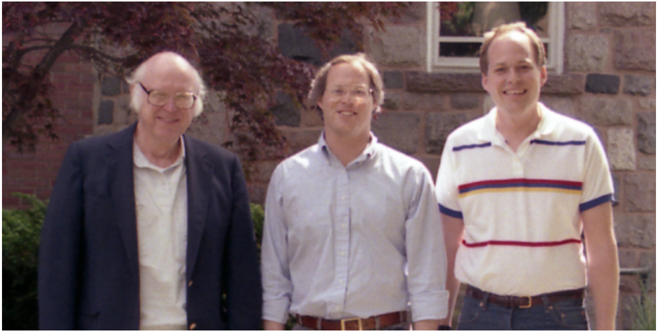
- Biological Modeling
  - does the architecture correspond to what we know about the brain?
  - Leabra, Nengo & SPA
- Psychological modeling
  - does the architecture capture the details of human performance?
  - ACT-R, Clarion
- Functionality
  - does the architecture explain how human intelligence?
  - does the architecture support creation of useful systems?
  - Soar, DIARC (cognitive robotics), Icarus

Common Assumptions

- Symbolic, relational representations
- Complex behavior arises from simple decisions controlled by knowledge
- Significant internal parallelism, limited external parallelism
- ~50 msec basic cycle time to achieve real-time behavior
- Agent is always *on* and must do all processing and learning while behaving
- Knowledge access is bounded to maintain reactivity
- Learning is incremental & on-line

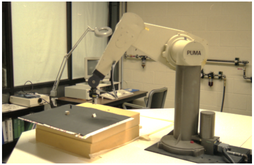
The Soar Cognitive Architecture
Laird 2012

- Multi-method, multi-task problem solving and learning
- Combines problem-spaces and production systems
- Inspired by human problem solving (*The United Theories of Cognition*, Newell 1991)



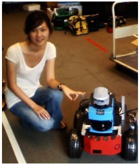
Soar on Robots

1988:Robo-Soar,UM

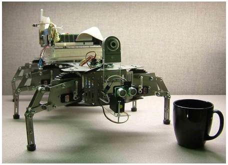


1990:Hero-Soar,UM

2004:Adapt,Pace U.



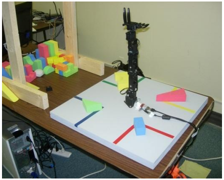
2009:Penn State U



2009:Splinter,UM

2010:SoarTech

2011:Magic-Soar,ST



2012-:Rosie,UM

Structural Overview

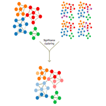
Sensors 

Actuators 

Structural Overview

Perception

classification, clustering, probabilistic representation

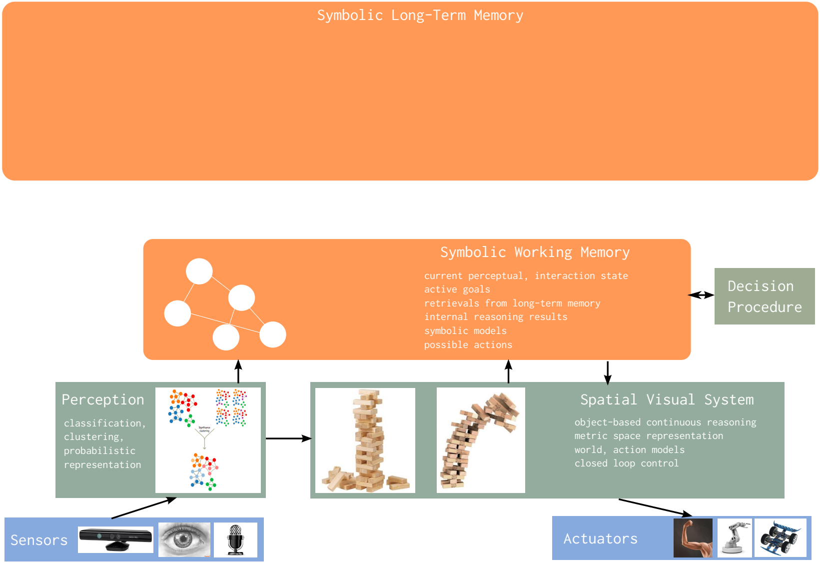


Sensors

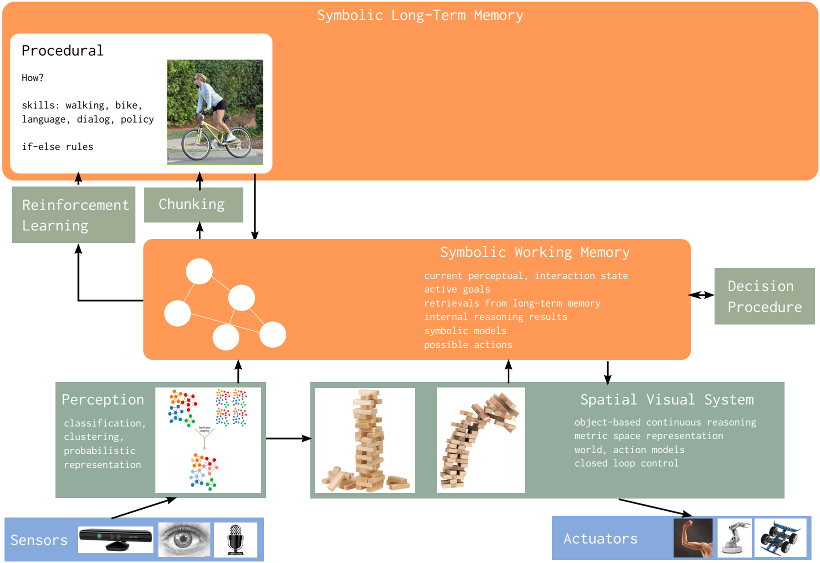
Actuators

Structural Overview

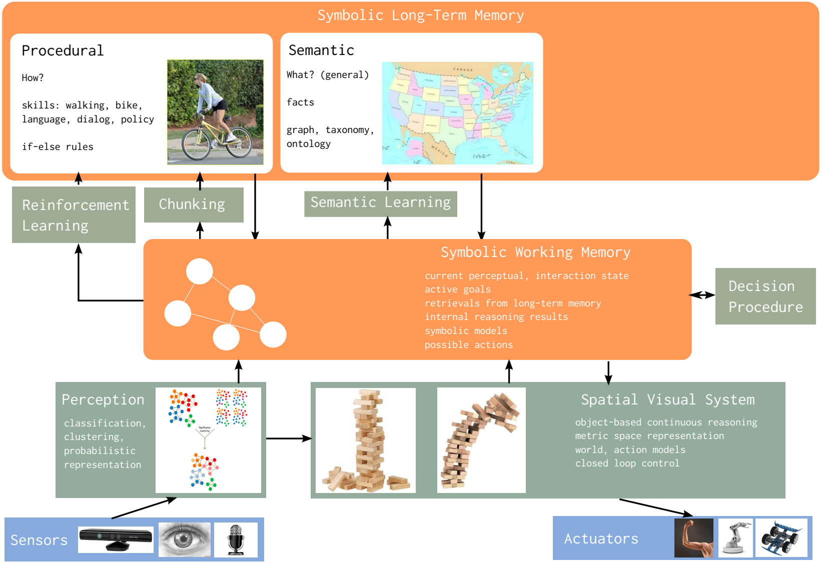
Structural Overview



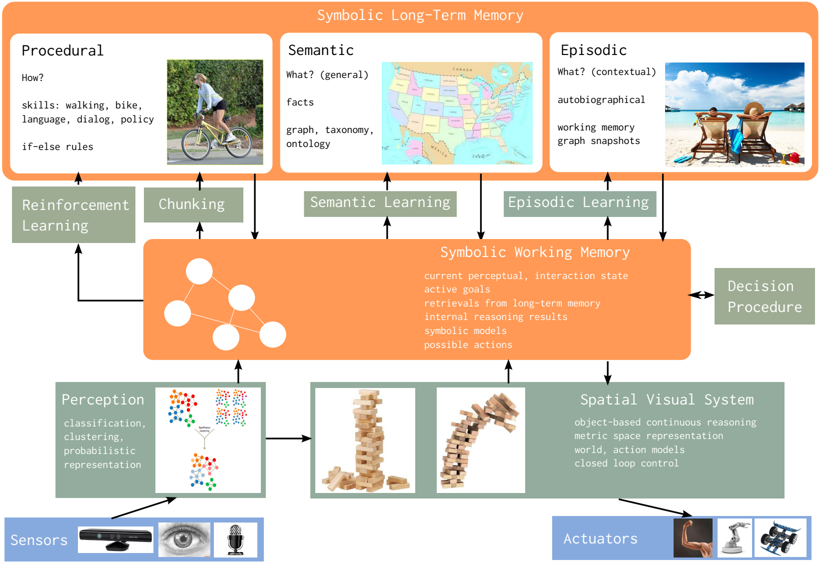
Structural Overview



Structural Overview

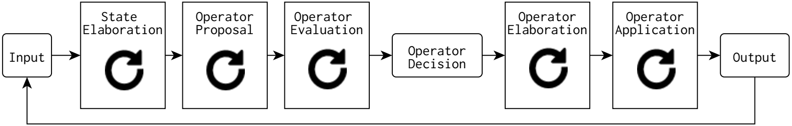


Structural Overview



Problem Space Computation Model (PSCM)

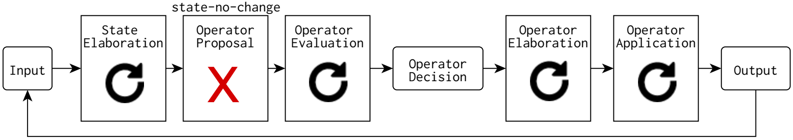
Locus of decision: operator selection (internal/external actions)



Claim: a *complete* model of reasoning

Missing/Conflicting Knowledge

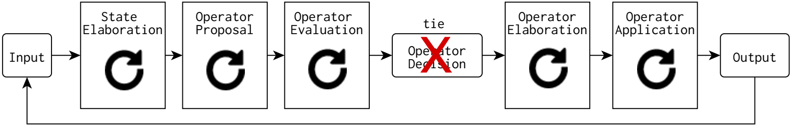
Impasse and substates



Impasses and substate reasoning drive interaction and learning in Rosie

Missing/Conflicting Knowledge

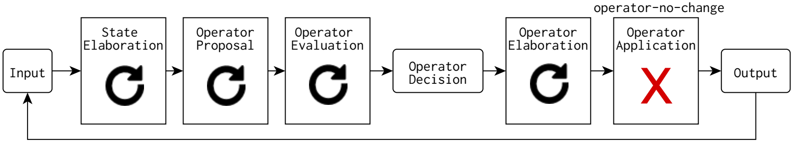
Impasse and substates



Impasses and substate reasoning drive interaction and learning in Rosie

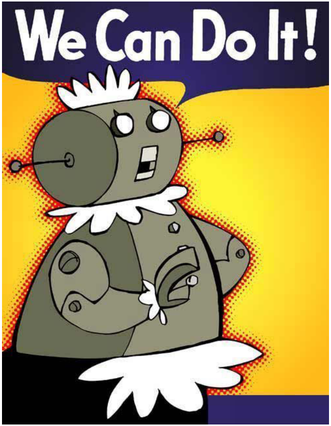
Missing/Conflicting Knowledge

Impasse and substates



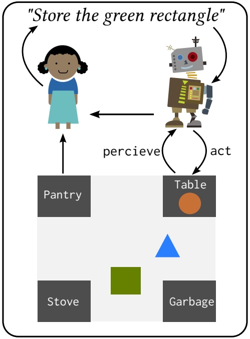
Impasses and substate reasoning drive interaction and learning in Rosie

Rosie: A Framework for Situated Interactive Instruction

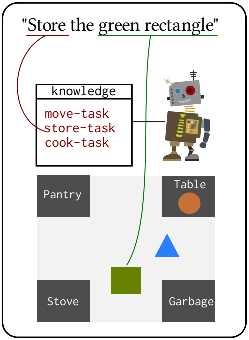


Overview

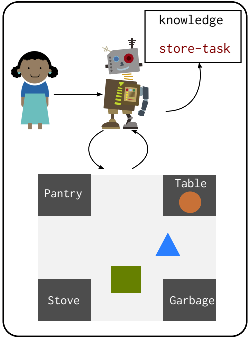
Integrative Interaction



Situated Comprehension



Task Learning



Task Knowledge

Given a set of primitive actions and their models
Acquire the following for `store`

What?

- parameters
  Store the green cylinder.
  {store(O2, pantry, in(O2,pantry))}
- subtasks
  `store`: `open`, `move [pick-up, put-down]`, `close`
- goal
  `in(O2,pantry)` ∧ `closed(pantry)`

How?

- policy
  if `[state,task]` then `execute([subtask])`
- model
  if `[state,task]` then `[next-state]`

When?

- availability
  if `[state]` then `available(store)`
- termination
  if `[state]` then `terminate(store)`

interactive Explanation-based Learning

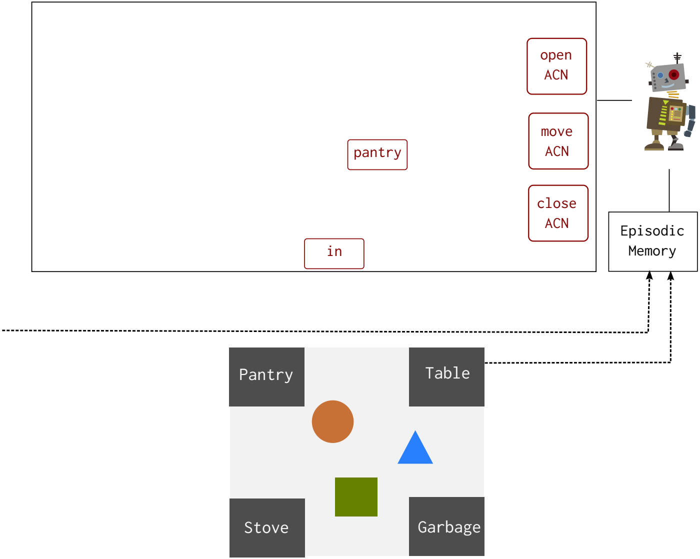
EBL methods: Mitchell 1986, Rosenbloom (1986)

Specific to general learning

1. (interactive) Acquire a specific example of how to execute a task.
2. (EBL) Generalize the specific experience

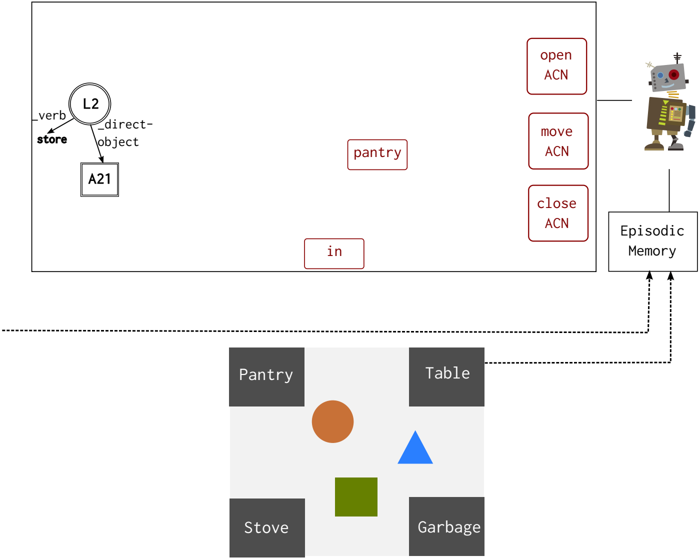
1. Interactive Example Execution

<u>Interaction trace</u>
Instructor: Store the green rectangle.



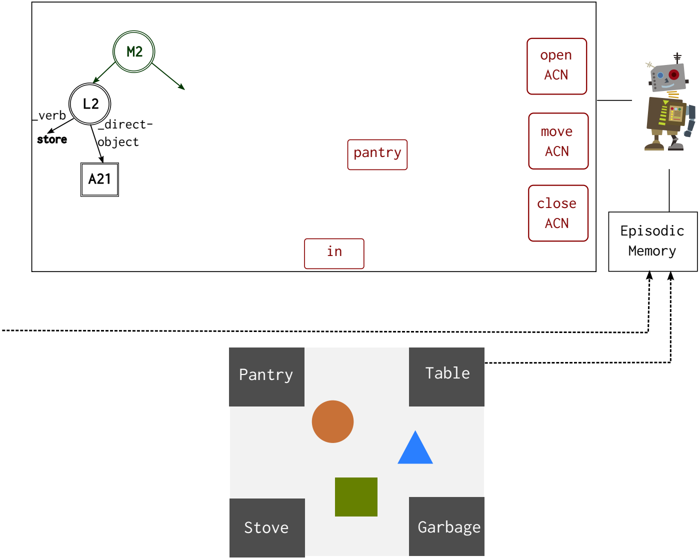
1. Interactive Example Execution

Interaction trace
Instructor: Store the green rectangle.



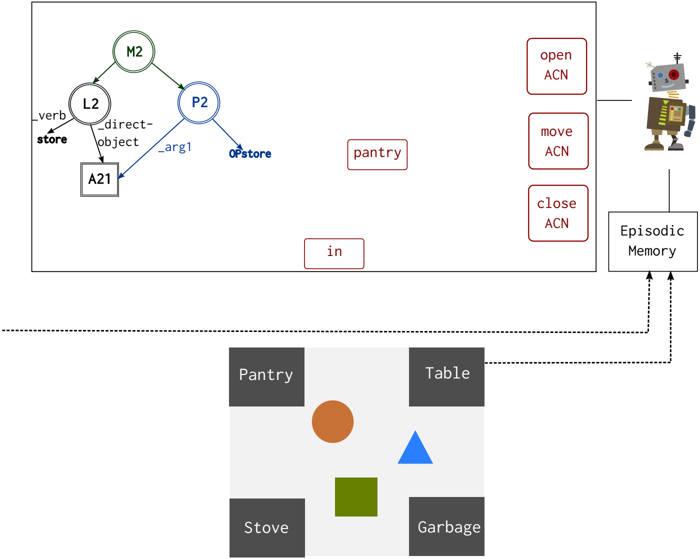
1. Interactive Example Execution

Interaction trace
Instructor: Store the green rectangle.



1. Interactive Example Execution

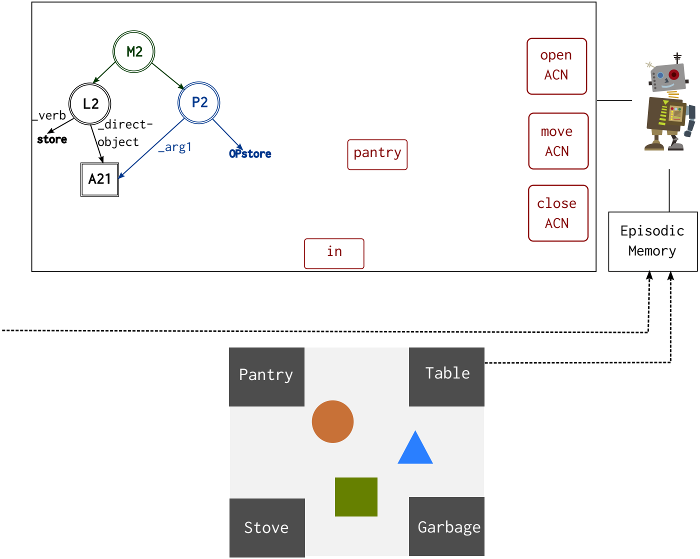
<u>Interaction trace</u>
Instructor: Store the green rectangle.



1. Interactive Example Execution

Interaction trace
Instructor: Store the green rectangle.

Agent: What is the goal of the action?



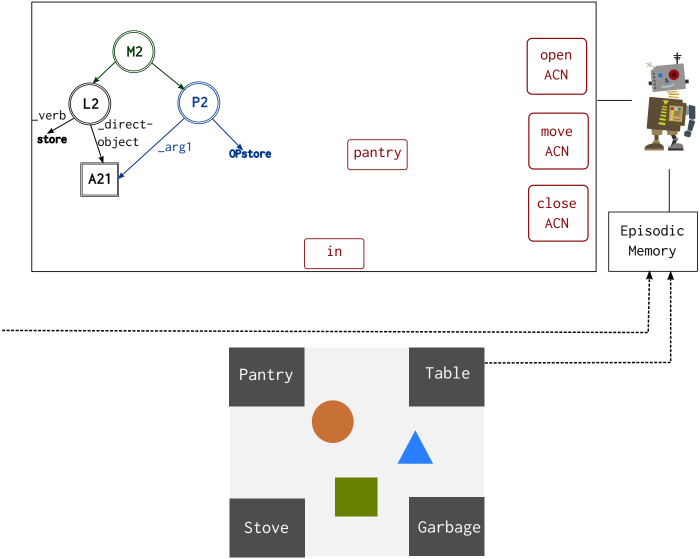
1. Interactive Example Execution

<u>Interaction trace</u>

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.



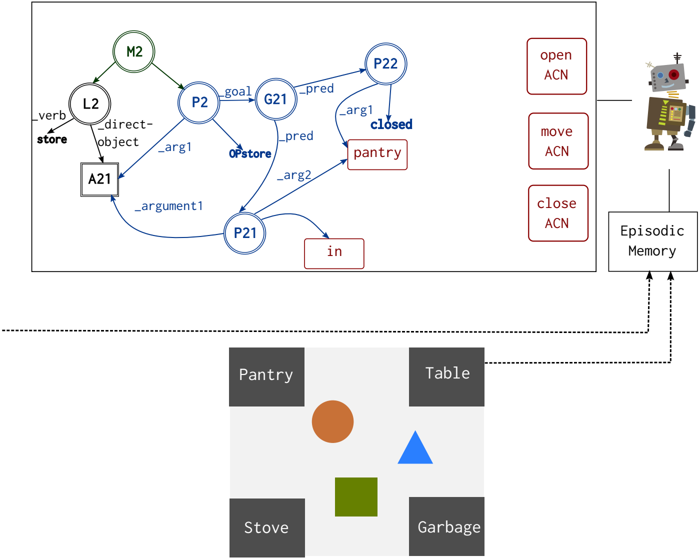
1. Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.



1. Interactive Example Execution

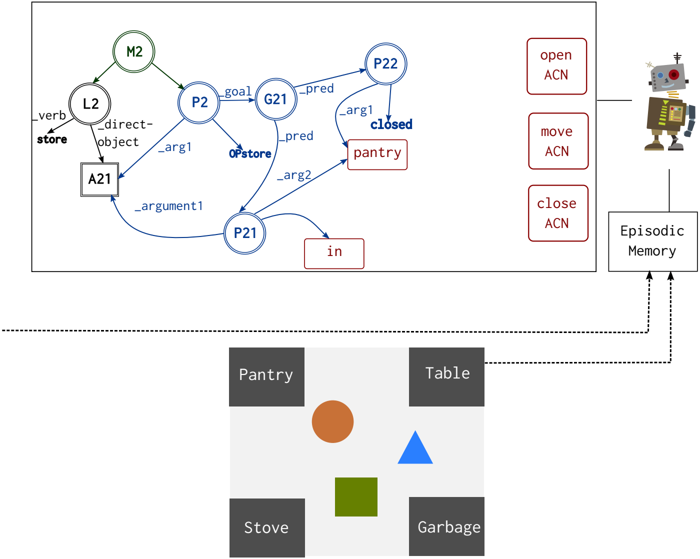
<u>Interaction trace</u>

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

Agent: Which action should I take?



1. Interactive Example Execution

<u>Interaction trace</u>

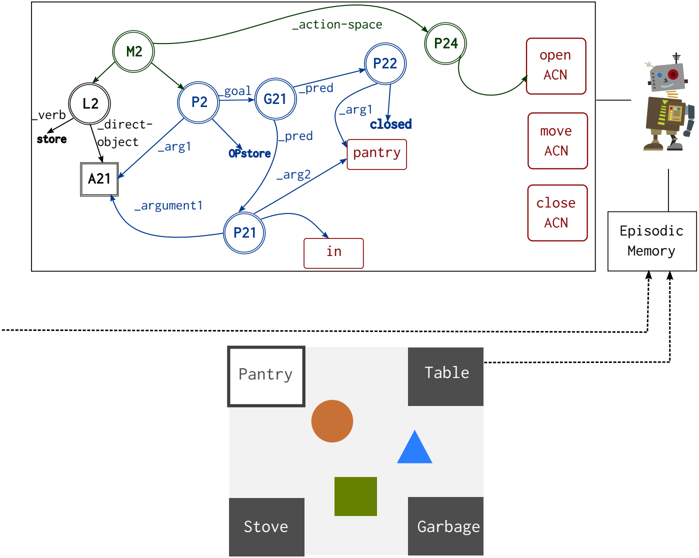
Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

Agent: Which action should I take?

Instructor: Open the pantry.



1. Interactive Example Execution

<u>Interaction trace</u>

Instructor: Store the green rectangle.

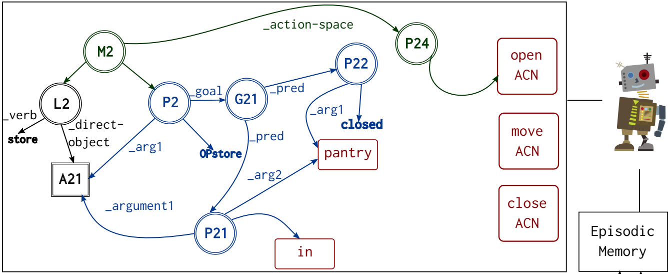
Agent: What is the goal of the action?

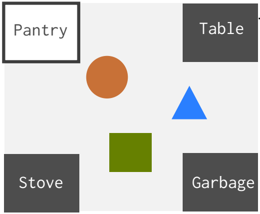
Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

Agent: Which action should I take?

Instructor: Open the pantry.

Agent: Which action should I take?





1. Interactive Example Execution

<u>Interaction trace</u>

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

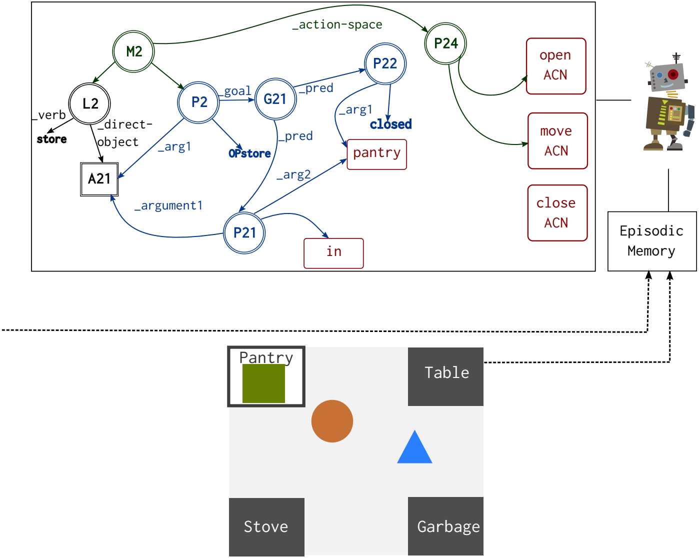
Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

Agent: Which action should I take?

Instructor: Open the pantry.

Agent: Which action should I take?

Instructor: Move the object to the pantry.



1. Interactive Example Execution

<u>Interaction trace</u>

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

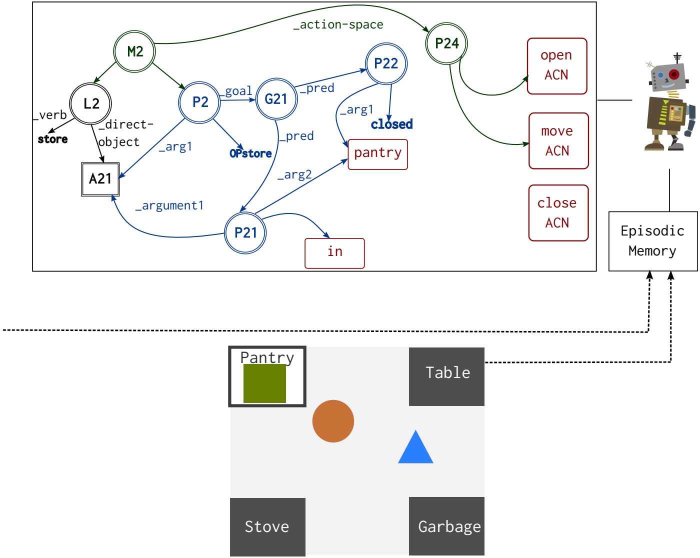
Agent: Which action should I take?

Instructor: Open the pantry.

Agent: Which action should I take?

Instructor: Move the object to the pantry.

Agent: Which action should I take?



1. Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

Agent: Which action should I take?

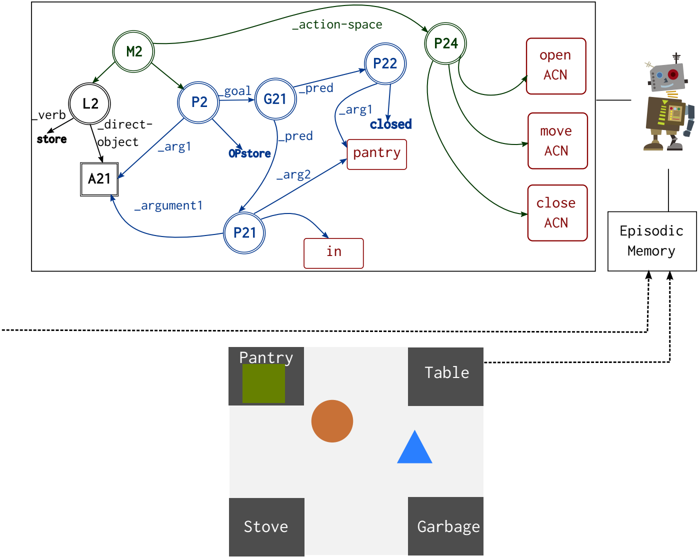
Instructor: Open the pantry.

Agent: Which action should I take?

Instructor: Move the object to the pantry.

Agent: Which action should I take?

Instructor: Close the pantry.



1. Interactive Example Execution

<u>Interaction trace</u>

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

Agent: Which action should I take?

Instructor: Open the pantry.

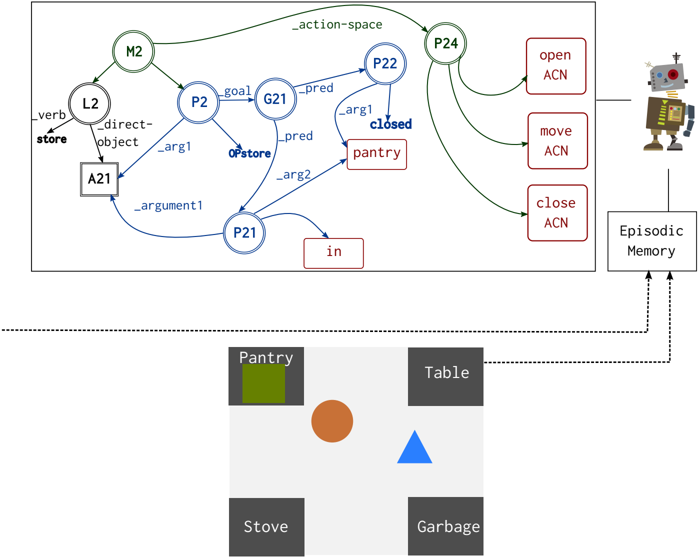
Agent: Which action should I take?

Instructor: Move the object to the pantry.

Agent: Which action should I take?

Instructor: Close the pantry.

Agent: Which action should I take?



1. Interactive Example Execution

<u>Interaction trace</u>

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

Agent: Which action should I take?

Instructor: Open the pantry.

Agent: Which action should I take?

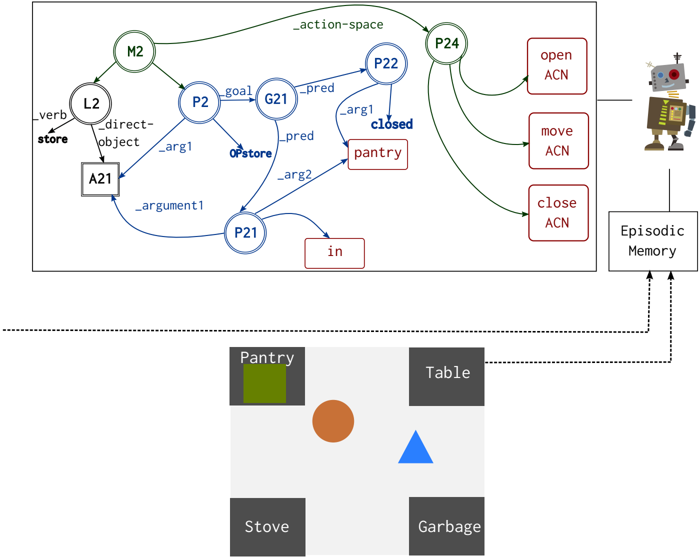
Instructor: Move the object to the pantry.

Agent: Which action should I take?

Instructor: Close the pantry.

Agent: Which action should I take?

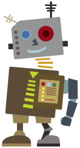
Instructor: You are done.



2. Retrospective Explanation

Concept Network

Episodic Memory

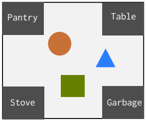


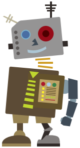
2. Retrospective Explanation

Concept Network

Episodic Memory

1. Recreate original state.



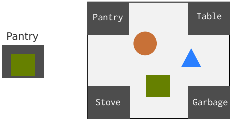


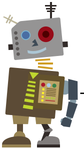
2. Retrospective Explanation

Concept Network

Episodic Memory

1. Recreate original state.
2. Generate desired state.



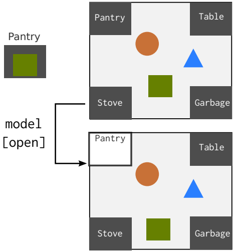


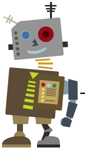
2. Retrospective Explanation

Concept Network

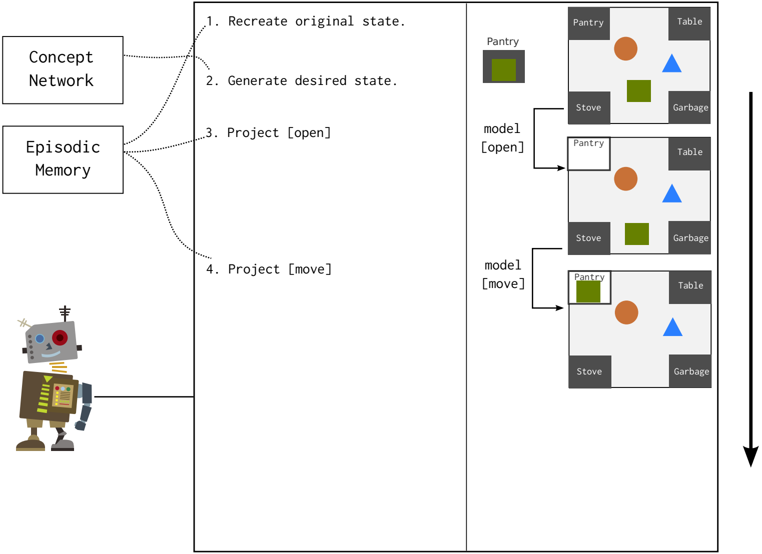
Episodic Memory

1. Recreate original state.
2. Generate desired state.
3. Project [open]

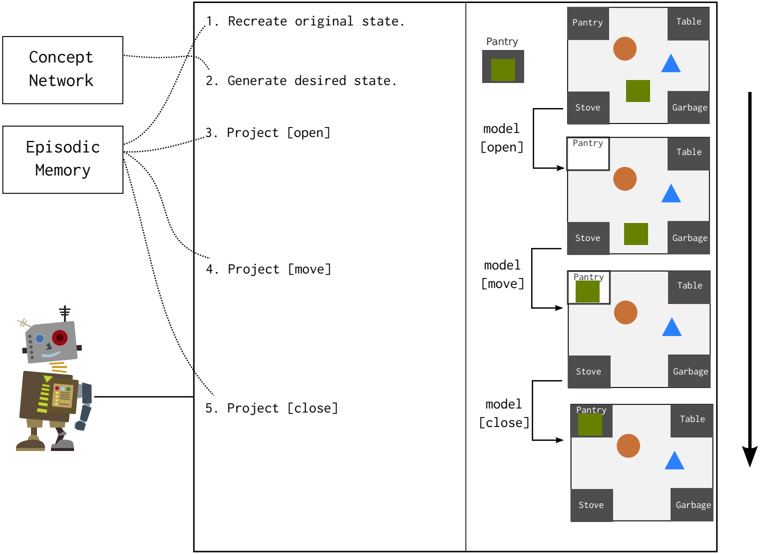




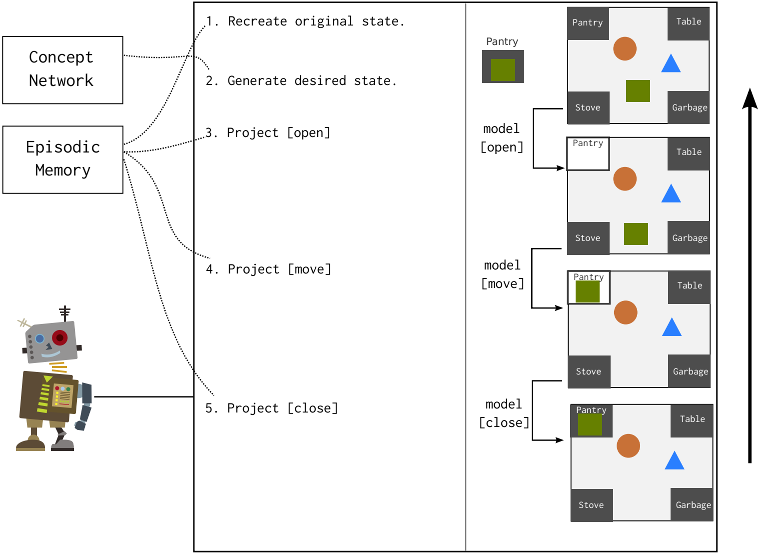
2. Retrospective Explanation



2. Retrospective Explanation

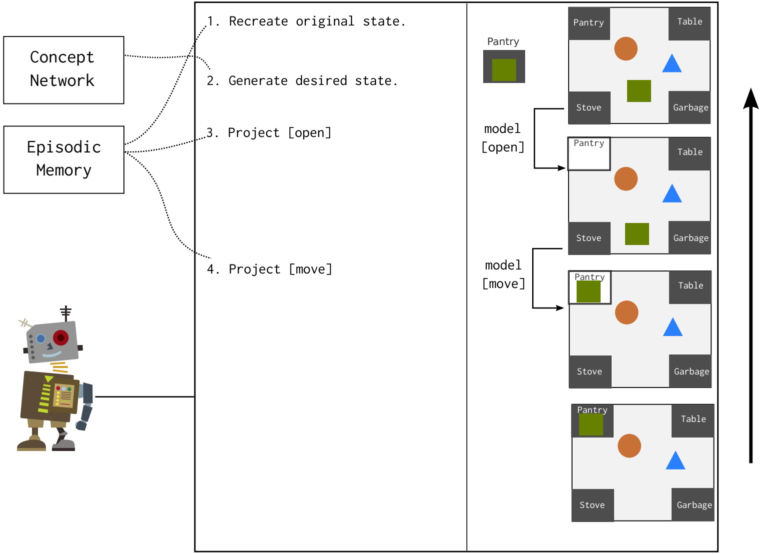


2. Retrospective Explanation



select close:
If `store([x])` and
`IN([x],PANTRY)` and
`OPEN(PANTRY)`
-->
select `close(PANTRY)`

2. Retrospective Explanation



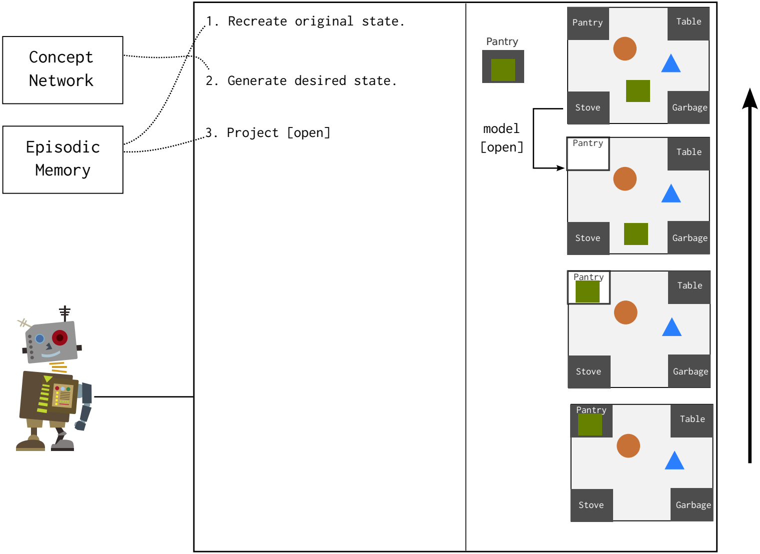
select close:
If `store([x])` and
`IN([x],PANTRY)` and
`OPEN(PANTRY)`
-->
select `close(PANTRY)`

---

select move:
If `store([x])` and
`-IN([x],PANTRY)` and
`OPEN(PANTRY)`
-->
select `move([x],PANTRY)`

---

2. Retrospective Explanation



select close:
If store([x]) and
IN([x],PANTRY) and
OPEN(PANTRY)
-->
select close(PANTRY)

select move:
If store([x]) and
-IN([x],PANTRY) and
OPEN(PANTRY)
-->
select move([x],PANTRY)

select open:
If store([x]) and
-IN([x],PANTRY) and
CLOSED(PANTRY)
-->
select open(PANTRY)

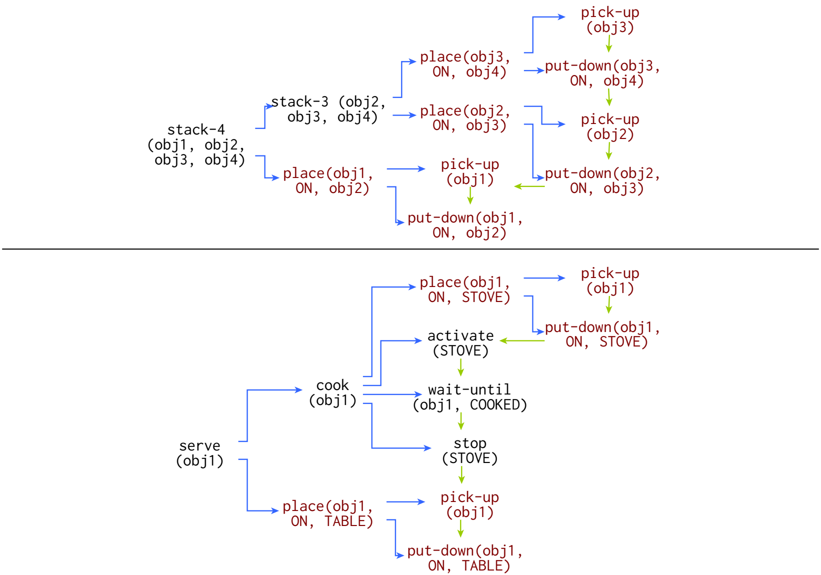
Comprehensive Learning

`place([x],[rel],[y])`, `move([x],[y])`, `discard([x])`

`store([x])`, `cook([x])`, `serve([x])`

`stack-3([x],[y],[z])`, `stack-4([x],[y],[z],[w])`

Hierarchical Transferable Representation



General Learning

Predicate selection

select open:
If `store(O1)` and `-IN(O1,PANTRY)` and `CLOSED(PANTRY)` and {`CLOSED(STOVE)` and `OFF(STOVE)` and `-ON(O2,STOVE)` and ...}
`-->`
select `open(PANTRY)`

Object variablization

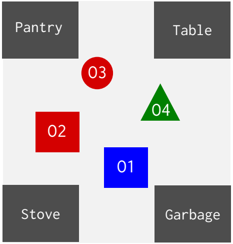
Store the green rectangle.
The goal is the green rectangle in the pantry and the pantry is closed.
Open the pantry.
Move the green rectangle to the pantry.

...

Situated Comprehension

- Communication
  - is situated
  - is contextual
    - efficient, effective
    - linguistically ambiguous
  - is interactive
- Comprehension model
  - be referential
  - exploit non-linguistic context
  - inform interaction
- Challenges
  - diverse, mixed representations
  - continual knowledge acquisition

Store the blue rectangle.
The goal is the rectangle is in the pantry.
...
Pick it up.
...



Hypothesis

- Linguistic communication is reference
- Speaker/hearer have a common ground
  - shared perceptions
  - common domain knowledge
  - similar experiences
- Linguistic features cues to search common ground
- Language specifies scene, knowledge fills up gaps

The Indexical Model

inspired by the Indexical Hypothesis: Glenberg and Robertson (1999)

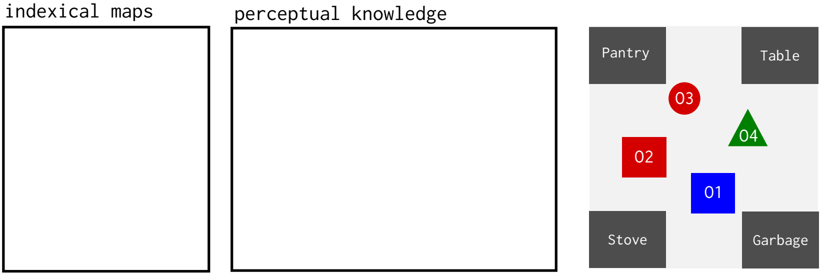
3 steps

1. Index words and phrases to referents
    - *NN/ADJ* $\rightarrow$ perceptual classification
    - *NP* $\rightarrow$ set of objects
    - ...
2. Extract domain-knowledge associated with referents
    - pre-encoded or learned (Mohan et al. 2012)
3. Mesh linguistic and syntactical constraints

The Indexical Model

Step 1: Index components

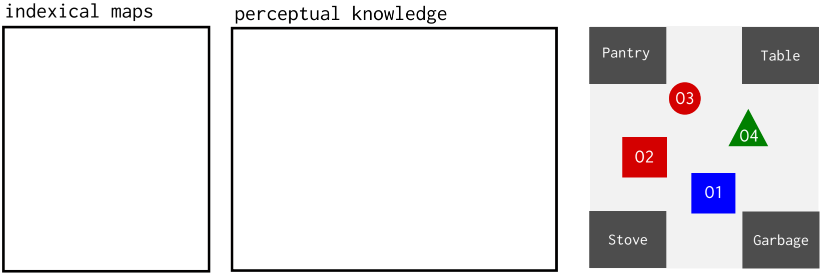
Move the blue object to the right of the pantry.



The Indexical Model

Step 1: Index components

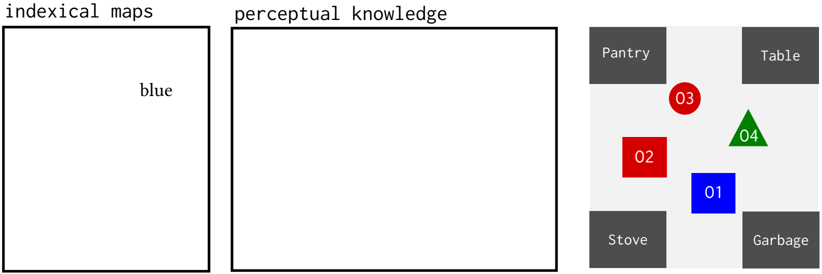
Move the blue object to the right of the pantry.



The Indexical Model

Step 1: Index components

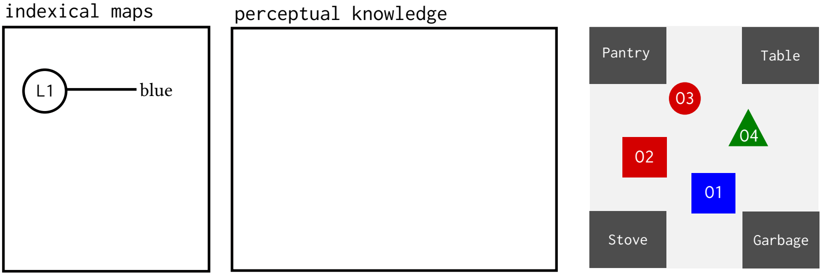
Move the blue object to the right of the pantry.



The Indexical Model

Step 1: Index components

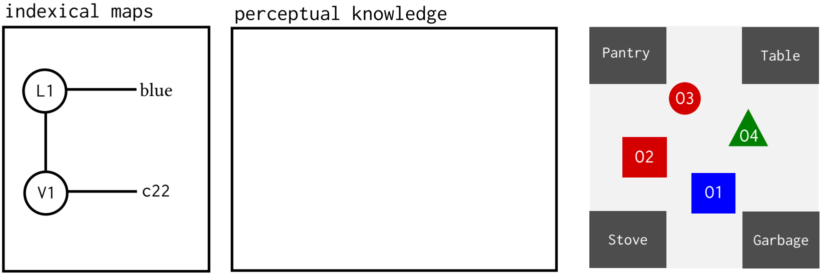
Move the blue object to the right of the pantry.



The Indexical Model

Step 1: Index components

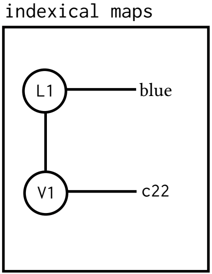
Move the blue object to the right of the pantry.

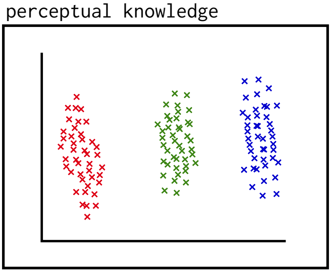


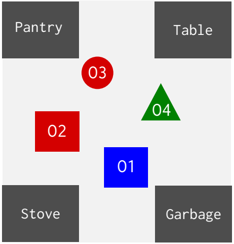
The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.



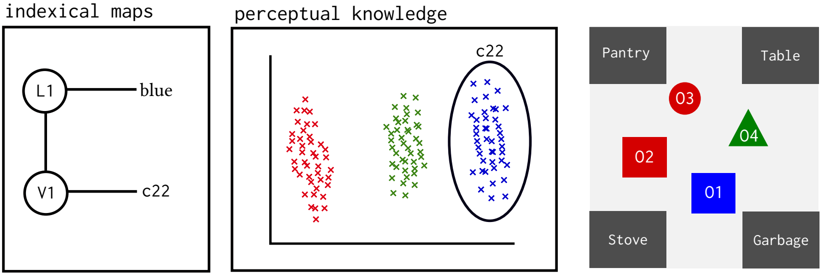




The Indexical Model

Step 1: Index components

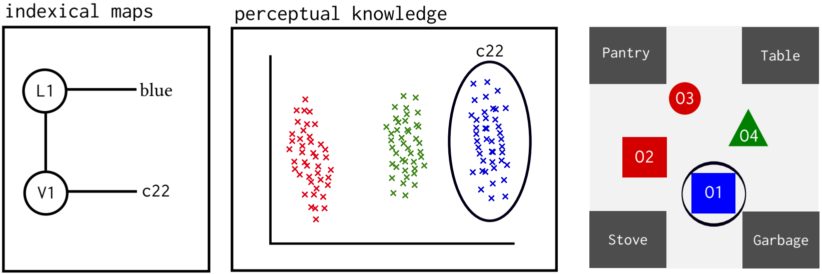
Move the blue object to the right of the pantry.



The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.



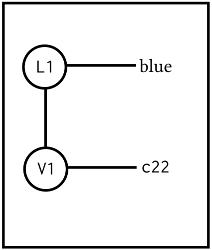
The Indexical Model

Step 1: Index components

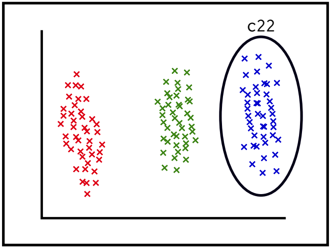
Move the blue object to the right of the pantry.

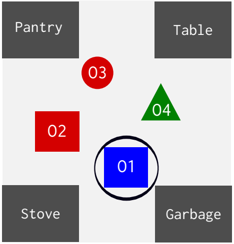
01

indexical maps



perceptual knowledge





The Indexical Model

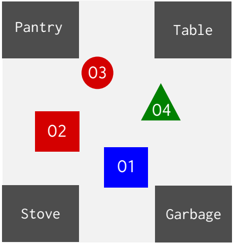
Step 1: Index components

Move the blue object to the right of the pantry.

O1

indexical maps

spatial knowledge

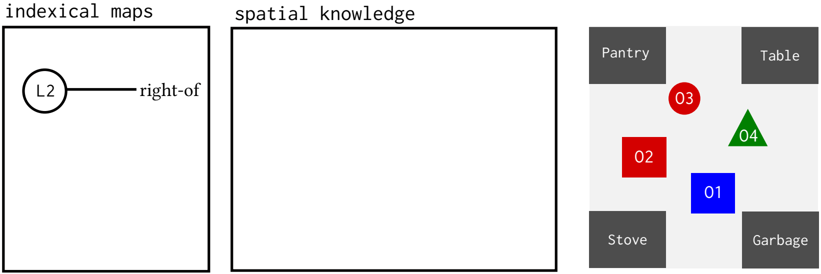


The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.

O1

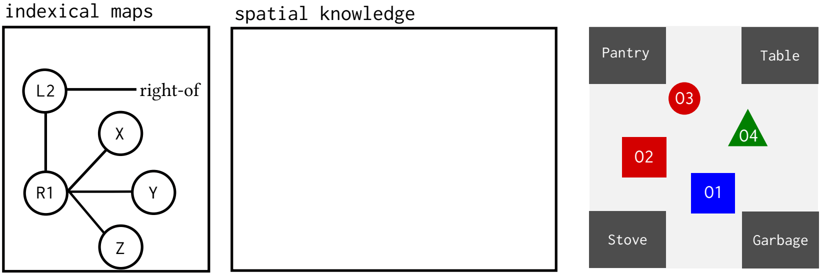


The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.

01

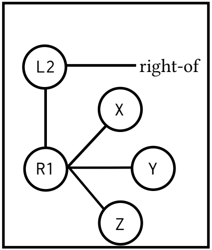


The Indexical Model

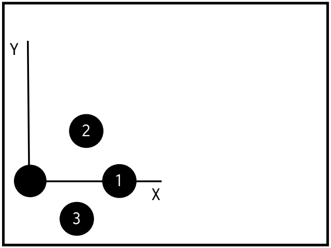
Step 1: Index components

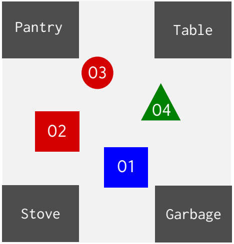
Move the blue object to the right of the pantry.
01

indexical maps



spatial knowledge





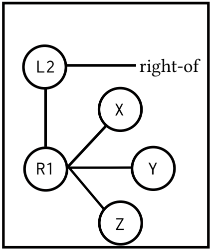
The Indexical Model

Step 1: Index components

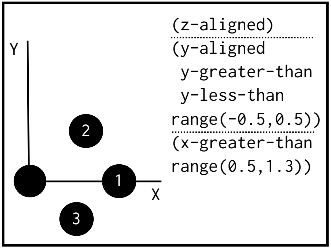
Move the blue object to the right of the pantry.

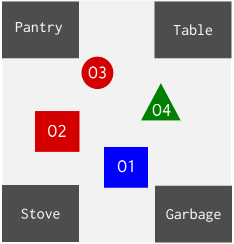
01

indexical maps



spatial knowledge





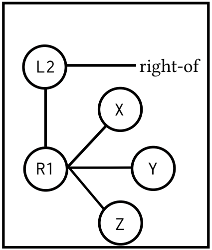
The Indexical Model

Step 1: Index components

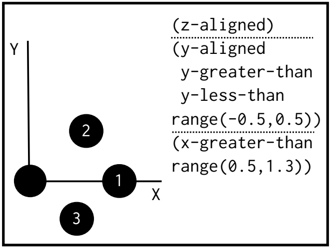
Move the blue object to the right of the pantry.

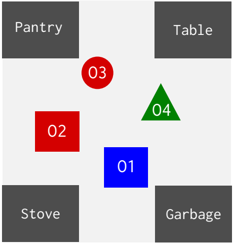
O1 R1

indexical maps



spatial knowledge

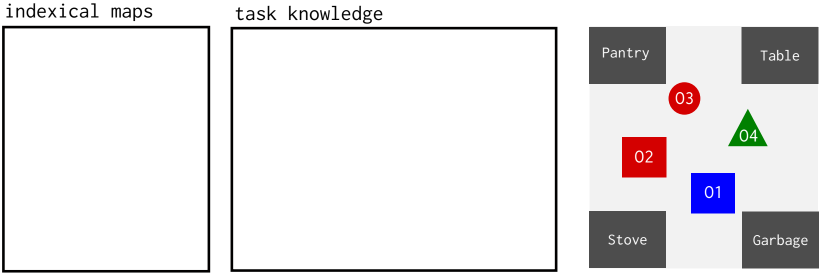




The Indexical Model

Step 1: Index components

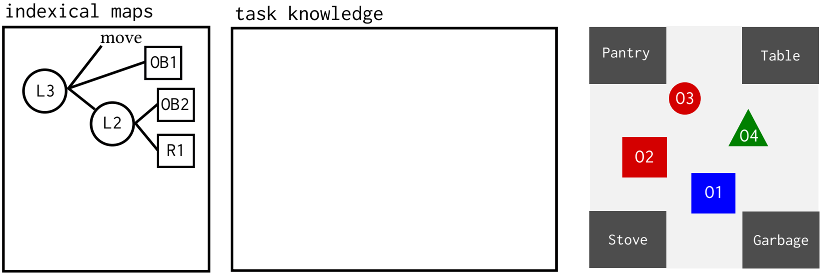
Move the blue object to the right of the pantry.
O1 R1



The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.
O1 R1

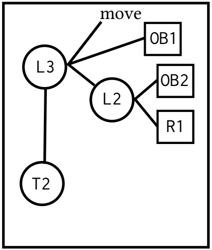


The Indexical Model

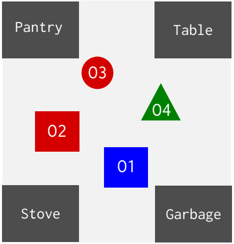
Step 1: Index components

<u>Move</u> the blue object to the right of the pantry.
O1 R1

indexical maps



task knowledge



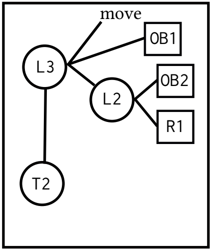
The Indexical Model

Step 1: Index components

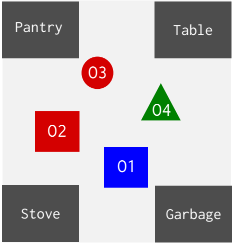
Move the blue object to the right of the pantry.

T2 O1 R1

indexical maps



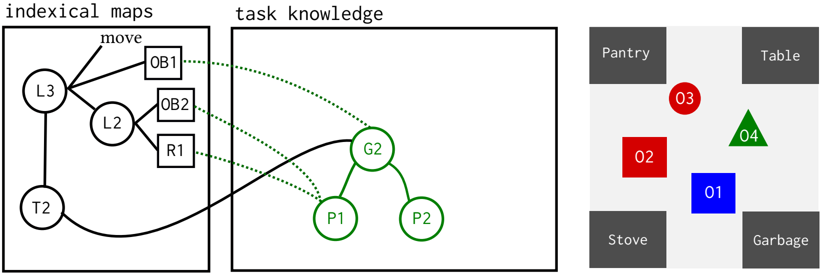
task knowledge



The Indexical Model

Step 1: Index components

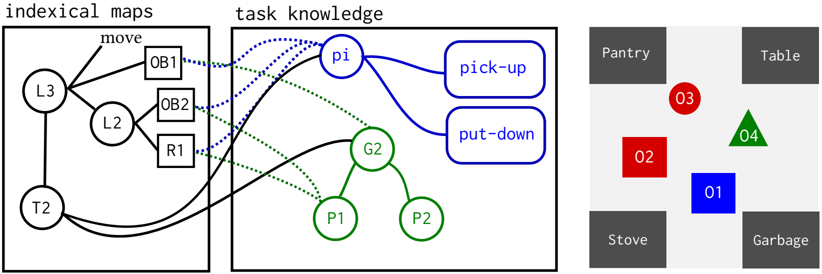
Move the blue object to the right of the pantry.
T2 O1 R1



The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.
T2 O1 R1

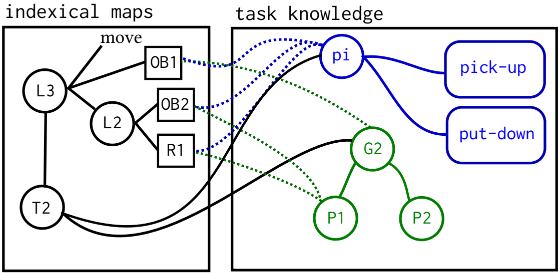


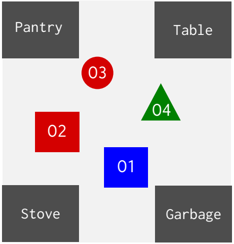
The Indexical Model

Step 2: Extract and instantiate domain knowledge

Move the blue object to the right of the pantry.

```
T2          O1                R1
task: T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
```





The Indexical Model

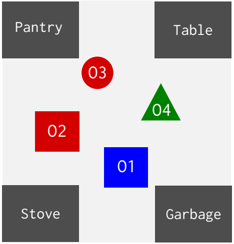
Step 3: Mesh constraints

Move the blue object to the right of the pantry.

```
T2          O1          R1
task: T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
```

```
available:
T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
T3(O1; G3(O1,(IN,pantry)); pi(O1,IN,pantry))
T4 ...
...


execute:
T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
```



Addressing Complexities

Natural language is ambiguous, does not specify complete information.
Non-linguistic context, knowledge provide information.

1. Referring Expressions
2. Unexpressed Verb Arguments

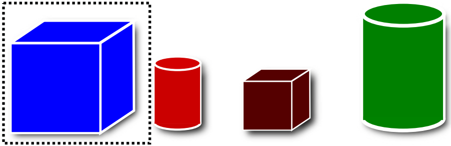
Referring Expressions

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube* }

Referring Expressions

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube* }

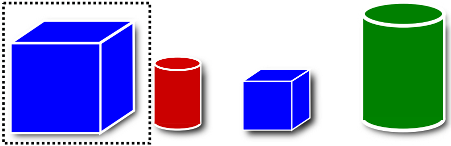
Pick up the blue cube.



Referring Expressions

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube* }

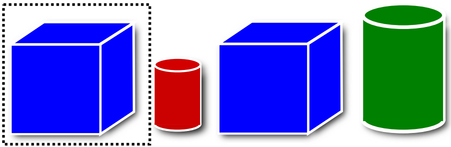
Pick up the large, blue cube.



Referring Expressions

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube*}

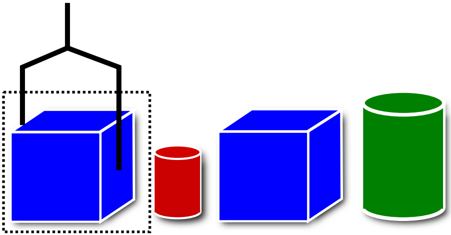
Pick up the cube on the left of the red cylinder.



Referring Expressions

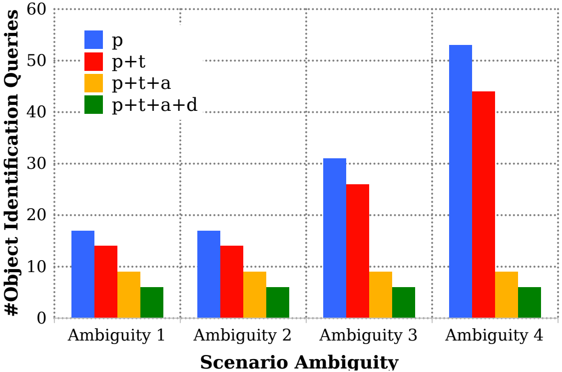
Referring expressions are situational {*it*, *this cube*, *that*, *the large cube* }

Put it down.



RE Resolution Performance

scenarios: number of distractors
models: *p*, *p+t*, *p+t+a*, *p+t+a+d*
corpus: instructional dialogs, 12 personal pronouns (*it*), 4 demonstrative pronouns (*this*), 3 demonstrative phrases (*that cylinder*), and 14 noun phrases (*the red cylinder*)



Stanford CoreNLP fails at 28.6% of references.

Games

Learning Tic-Tac-Toe
Playing Tic-Tac-Toe

Summary

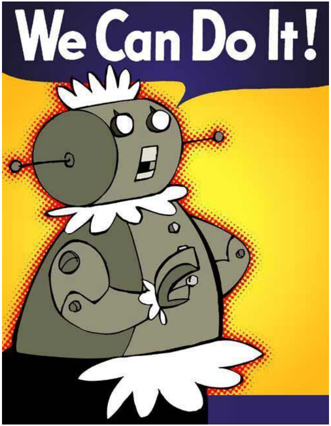
- Interaction is useful for learning
- Situated Interactive Instruction (SII)
  - *knowledge-level* interactions
  - natural language refers to concepts known to the agent
  - instructor can compose new knowledge
- SII may help task learning
  - generalization: identifying useful features (predicates), variablization
  - transfer: tasks may have common substructure

Challenges

Integration of speech with dialog

- disfluencies: *uh*, *um*
  - communicative function?
- content
  - complete bottom-up/top-down belief propagation
- emotion
  - convey encouragement, disappointment
- multi-modal
  - gaze, gestures

Questions?



Thanks to:
Soar: John Laird, Aaron Mininger, James Kirk,
APRIL: Edwin Olson, Robert Goeddel, Lauren Hinkle