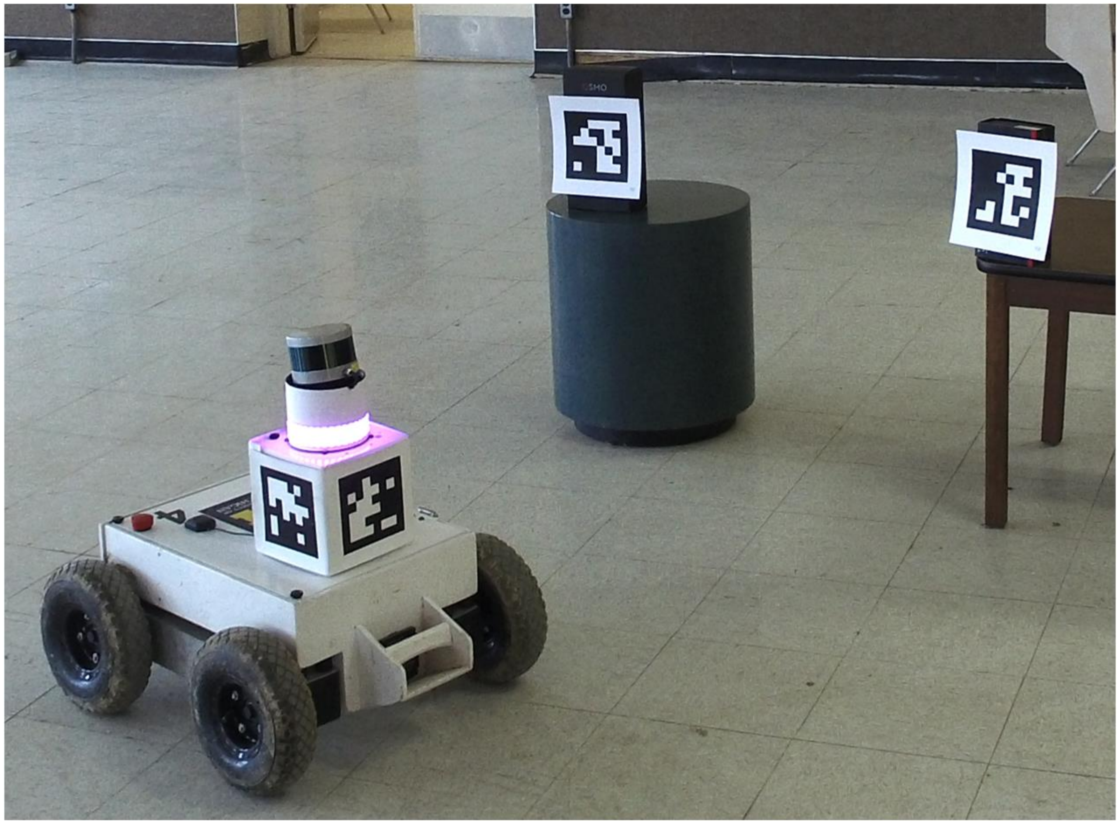
Interactively Learning Strategies For Finding And Anchoring Objects

Aaron Mininger, John Laird

Accepted Paper, ACS 2016

2016 Soar Workshop

Extending ITL to Mobile Domain



Research Question

Given a linguistic reference to an object, how does the agent connect that to a real object in the world?

Challenges:

- Partial Observability
- Novel Concepts
- Useful Knowledge in Different Memories

Research Question

Given a linguistic reference to an object, how does the agent connect that to a real object in the world?

Stages:

- □ Reference Resolution
- □ Object Finding
- □ Anchoring

Reference Resolution

Resolving an object reference to either an existing representation or a newly created one

Form	Level	Candidates
it	In-focus	Dialog
this	Activated	Gestured
the	Uniquely identifiable	Dialog > Visible > STM > LTM > NEW
a	Type identifiable	NEW

Object Finding

The agent must take actions so that a suitable object is perceived

Different strategy categories:

- Working Memory
- Long Term Memories
- Interaction
- External Search

Working Memory

- Knowledge about the object may be in working memory
  - Known Position: *Face the object*
  - Known Location: *Go to the kitchen*

- Knowledge may be added by other strategies

Long Term Memory: Semantic

- Semantic Memory may contain a useful fact for locating the object

- Created action *think*

  “Think of the <property> of <obj1>”

  “Think of the office of Bob”

  “This of the storage location of a soda”

Cue: **`(<obj1> ^<property> <obj2>)`**

Adds: **`in(<obj1>, <obj2>)`**

Long Term Memory: Episodic

- Information about a previous encounter with object may be in the agent's episodic memory

- Created action *recall*
  - Creates epmem cue based on description in command
  - Copies new information into working memory

- *Recall the soda in a location*

  cue: **`soda(A), location(B), in(A, B)`**
- *Recall Bob in an office*

Interaction

“I can’t find the package, can you help?”

- Instructor has many options
  - Tell location: “*The package is in the main office*”
  - Give a command: “*Turn around*”
  - Get the object herself: “*Here it is”*
  - Not do anything: “*I don’t know”*

External Search

- Perform actions to move around until a suitable object is visible
  - *Scan the room until you see the blue block*.
  - *Explore until you see a stapler*

Object Finding Subtask

Created `find(obj)` as a subtask

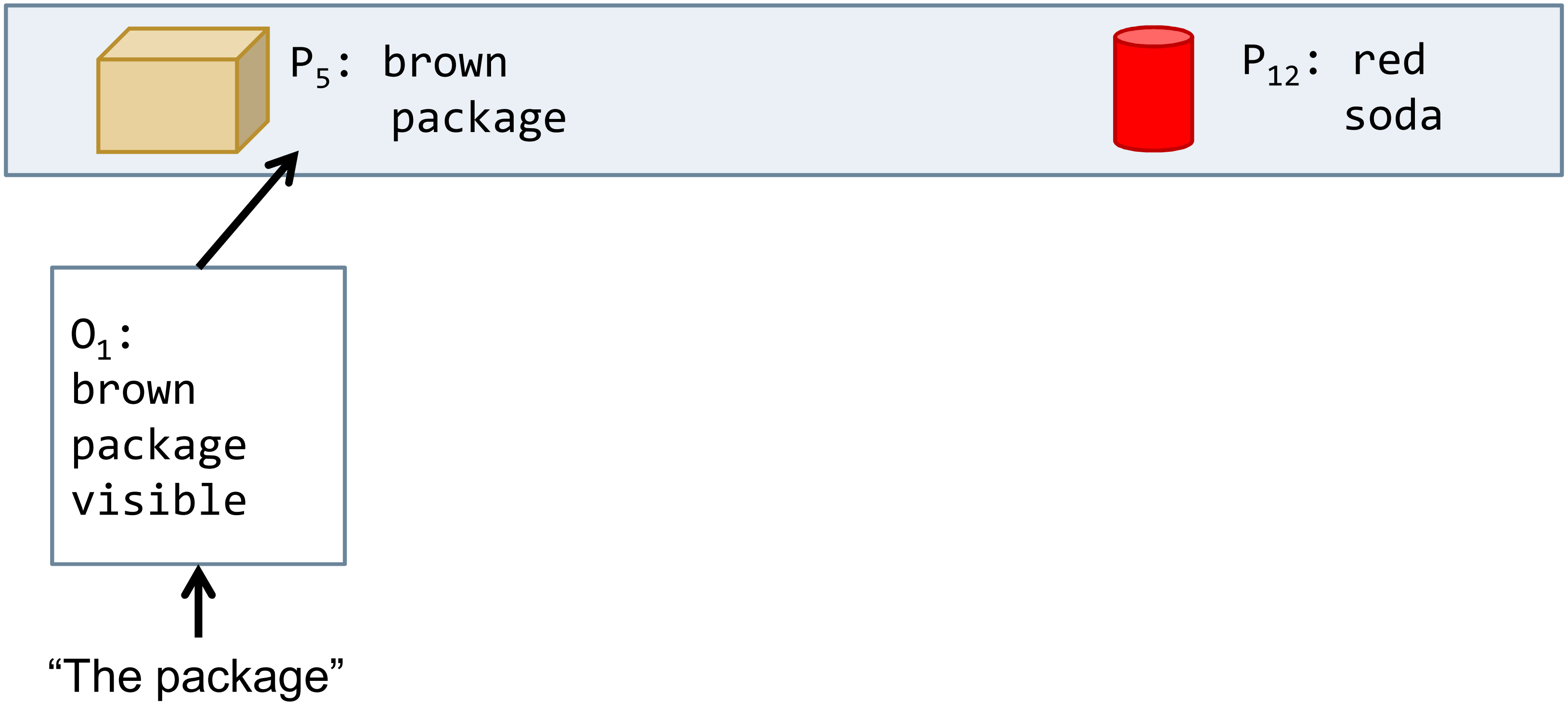
- Uses the `visible(obj)` predicate
- Can plan *when* to try and find an object
- Can learn strategies that transfer
- Preference order based on source:
  - STM > LTM > Interaction > External Search

Anchoring

The agent must connect an internal representation to its perception

Anchoring

Case 1: Object is already anchored



P_5: brown package

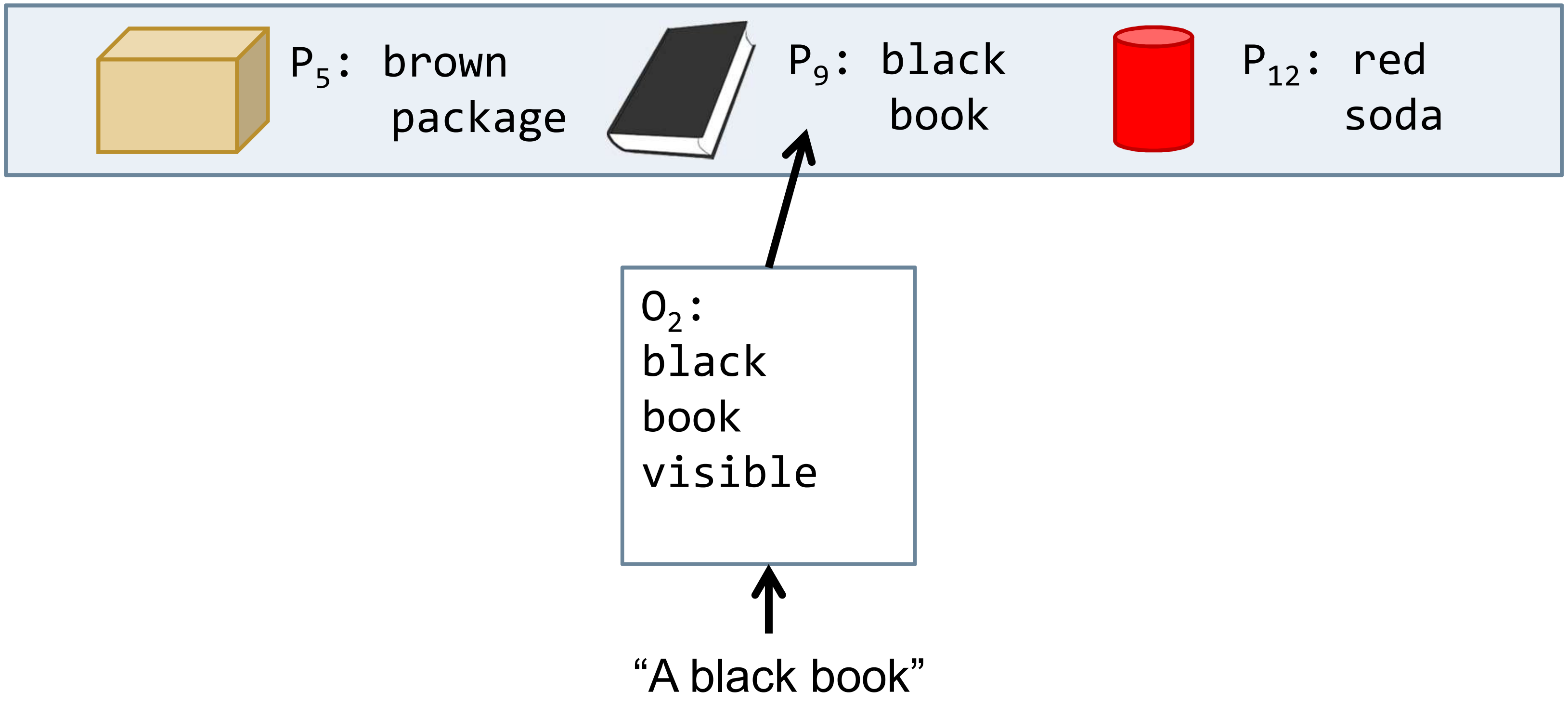
P_{12}: red soda

O_1: brown package visible

“The package”

Anchoring

Case 2: Anchoring is delayed



P_5: brown package

P_9: black book

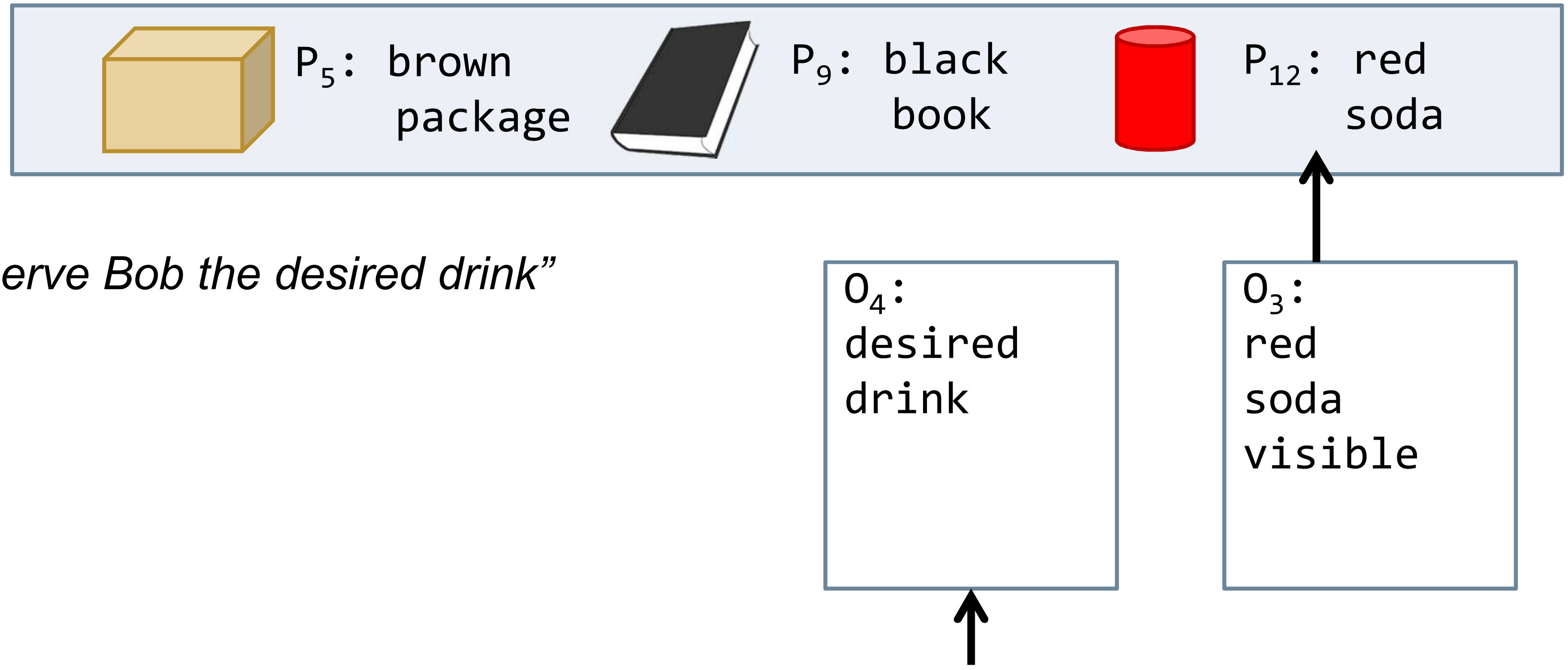
P_{12}: red soda

O_2:
black
book
visible

“A black book”

Anchoring

Case 3: Agent needs to learn how to anchor the reference



P_5: brown package

P_9: black book

P_{12}: red soda

“Serve Bob the desired drink”

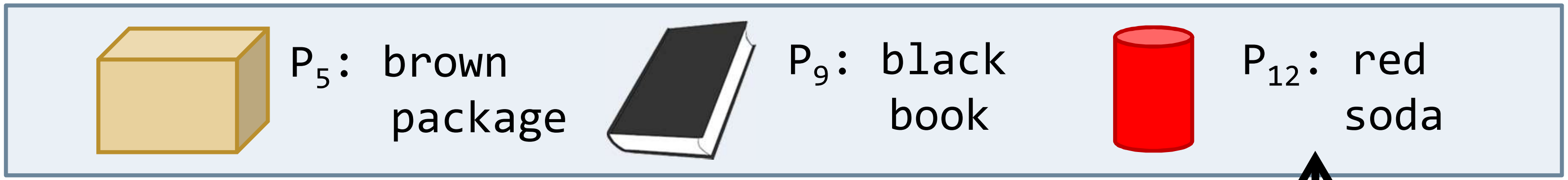
O_4: desired drink

O_3: red soda visible

“The desired drink”

Anchoring

The agent must connect an internal representation to its perception

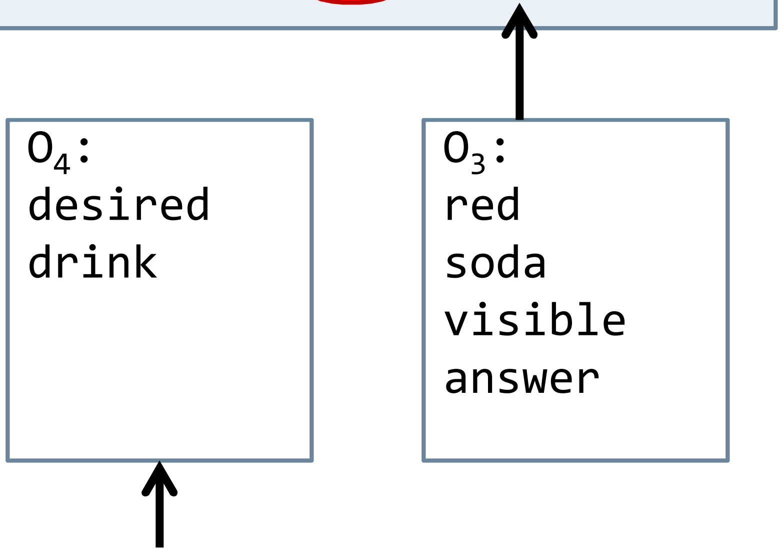


“Ask ‘What would you like to drink?’”

Rosie: “*What would you like to drink?*”

“The soda”

Rosie: “*What do I do next?*”



“The desired drink”

Anchoring

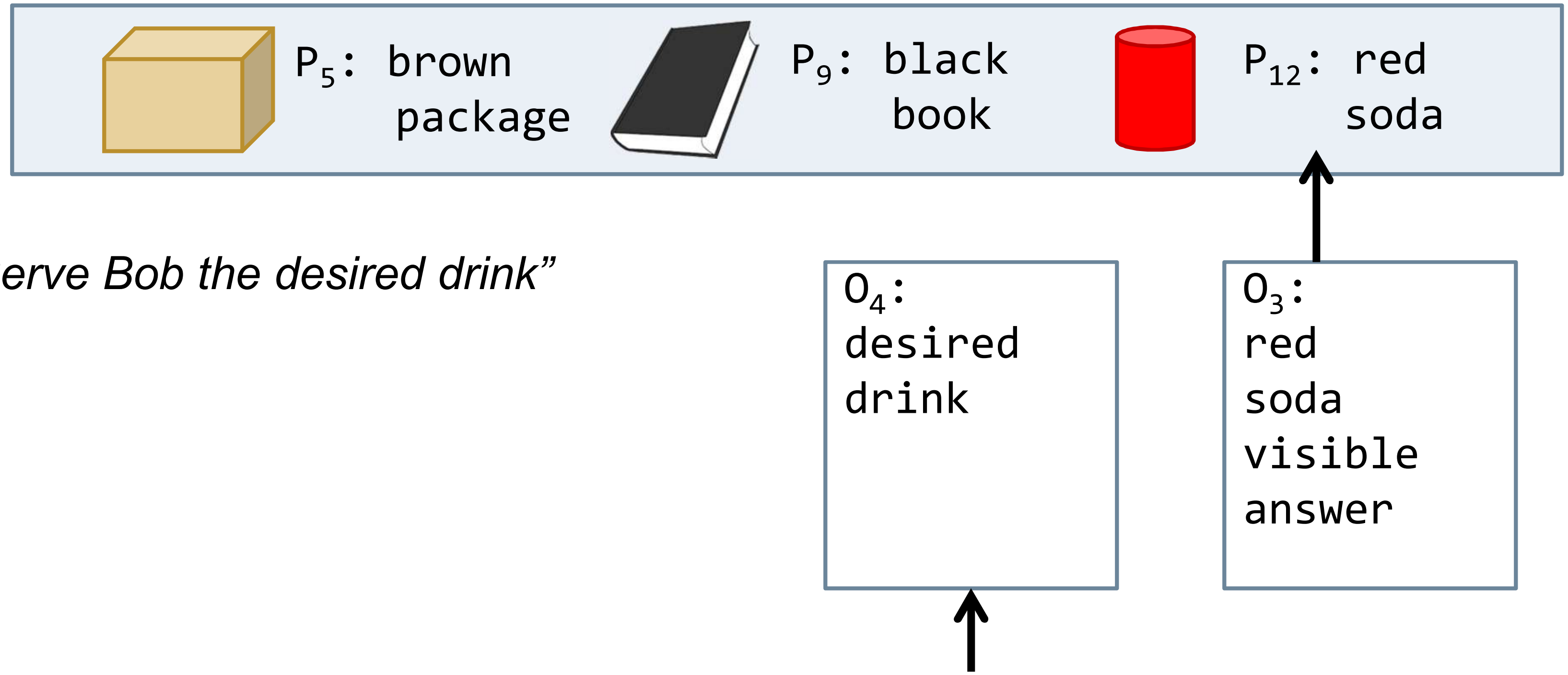
- Created action *remember*
  - Merges an anchored and un-anchored object

- *Remember the answer as the desired drink*

```
remember(A, as(B))
```

Anchoring

The agent must connect an internal representation to its perception



P_5: brown package

P_9: black book

P_{12}: red soda

“Serve Bob the desired drink”

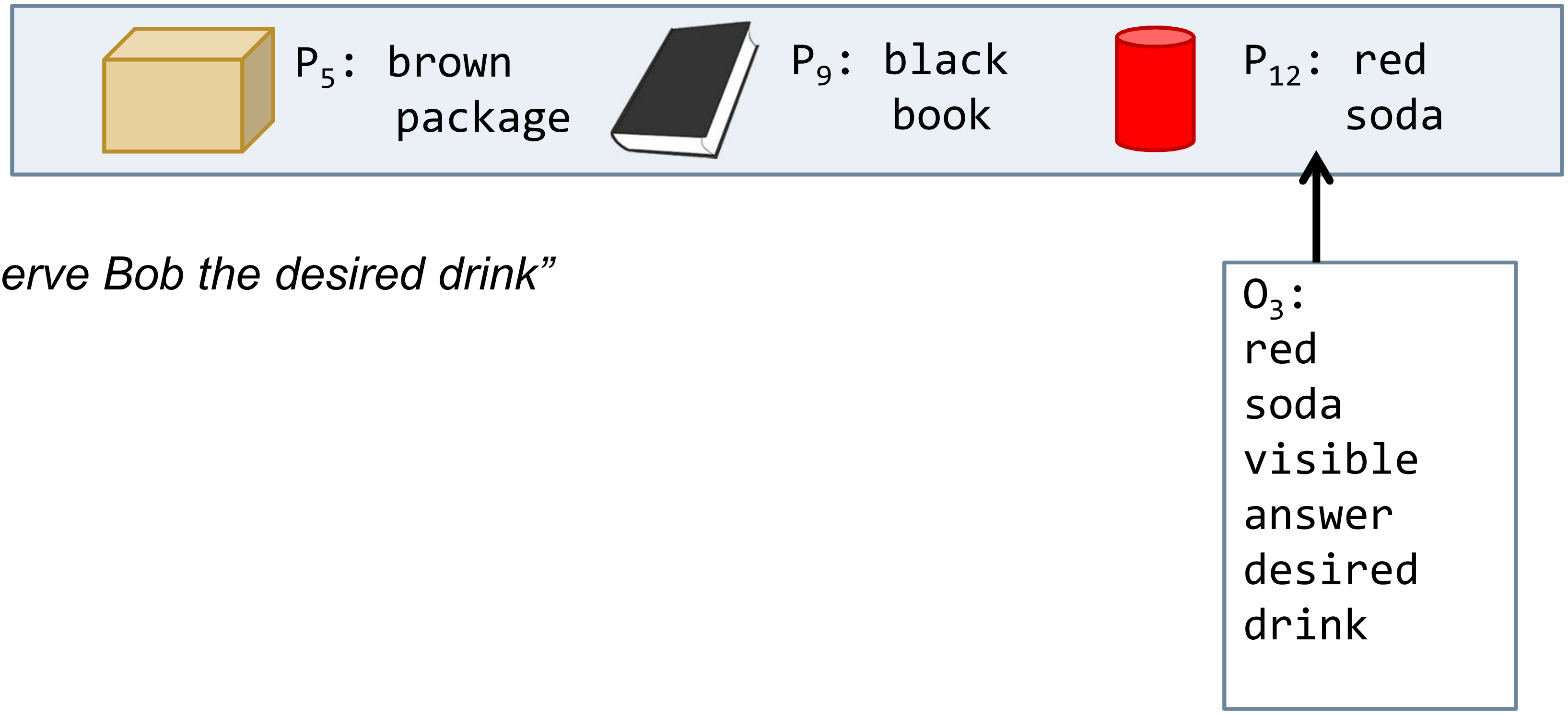
O_4: desired drink

O_3: red soda visible answer

“The desired drink”

Anchoring

The agent must connect an internal representation to its perception



P_5: brown package

P_9: black book

P_{12}: red soda

O_3: red soda visible answer desired drink

“Serve Bob the desired drink”

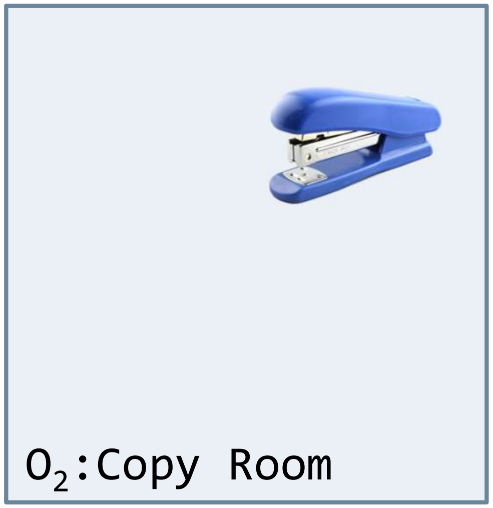
“The desired drink”

Example: Fetch a stapler

“Fetch a stapler”

‘a stapler’ [*type id*] -> NEW O_3

fetch(O_3)



Working Memory

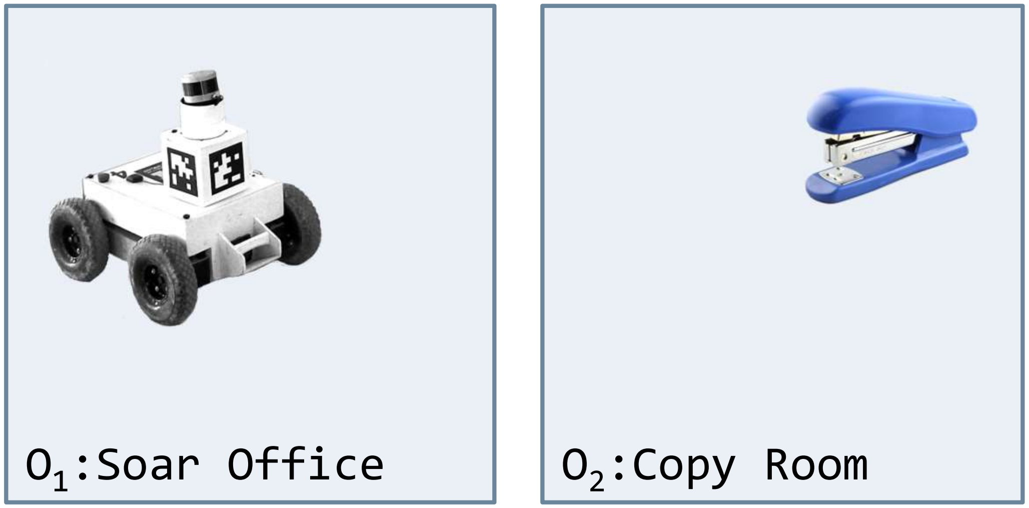
O_1: {soar, office,
current, location}

O_3: {stapler}

Task: fetch(O_3)

Example: Fetch a stapler

Rosie: *“What is the goal?”*



O_1:Soar Office

O_2:Copy Room

Working Memory

O_1: {soar, office,
current, location}

O_3: {stapler}

Task: fetch(O_3)

Example: Fetch a stapler

“The goal is that the stapler is in the starting location”

‘the stapler’ [*uniquely id.]* -> Dialog O_3

‘the starting location’ [*uniquely id*.] -> NEW O_4

O_1:Soar Office

O_2:Copy Room

Working Memory

O_1: {soar, office,
      current, location}

O_3: {stapler}

Task: fetch(O_3)

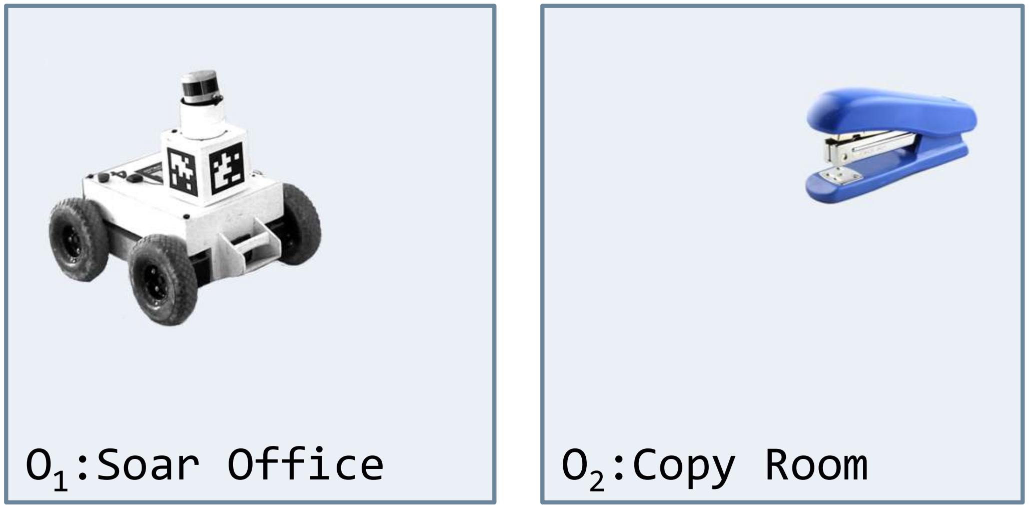
Example: Fetch a stapler

"The goal is that the stapler is in the starting location"

'the stapler' [*uniquely id.]* -> Dialog O_3

'the starting location' [*uniquely id*.] -> NEW O_4

Goal: $\mathtt{in}(O_3,\ O_4)$



Working Memory

O_1: {soar, office,

current, location}

O_3: {stapler}

O_4: {starting, location}

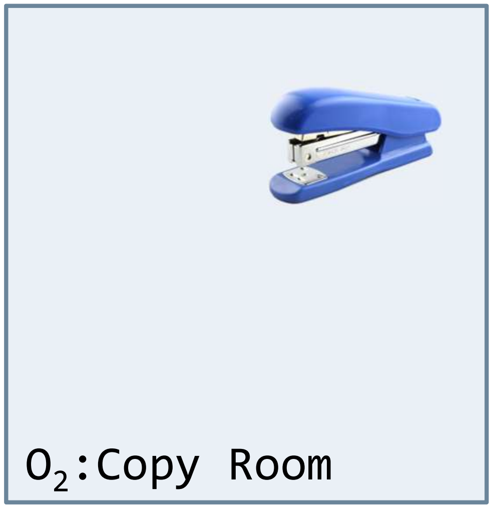
Task: fetch(O_3)

Goal: in(O_3, O_4)

Example: Fetch a stapler

Rosie: *“What do I do next?”*

O_1:Soar Office



Working Memory

O_1: {soar, office,
      current, location}

O_3: {stapler}

O_4: {starting, location}

Task: fetch(O_3)

Goal: in(O_3, O_4)

Example: Fetch a stapler

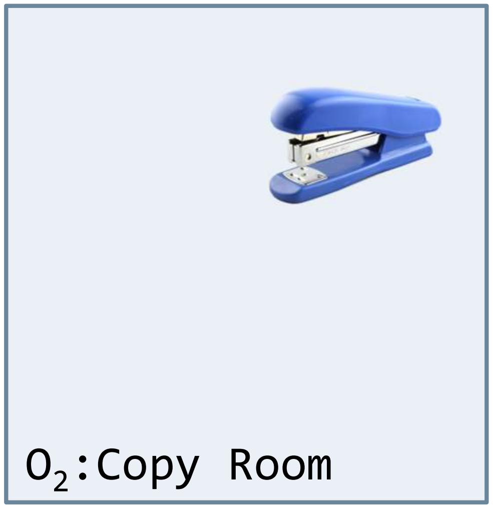
“Remember the current location as the starting location”

‘the current location’ [*uniquely id.]* -> WM O_1

‘the starting location’ [*uniquely id*.] -> Dialog O_4

remember(O_1, as(O_4))

O_1:Soar Office



O_2:Copy Room

Working Memory
O_1: {soar, office,
      current, location}
O_3: {stapler}
O_4: {starting, location}

Task: fetch(O_3)
Goal: in(O_3, O_4)

Example: Fetch a stapler

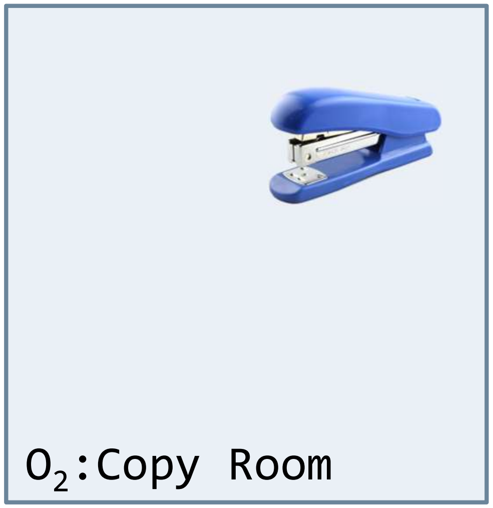
“Remember the current location as the starting location”

‘the current location’ [*uniquely id.]* -> WM O_1

‘the starting location’ [*uniquely id*.] -> Dialog O_4

remember(O_1, as(O_4))

O_1:Soar Office


O_2:Copy Room

Working Memory
O_1: {soar, office,
current, location,
starting}
O_3: {stapler}

Task: fetch(O_3)
Goal: in(O_3, O_1)

Example: Fetch a stapler

Action	Preconditions	Postconditions
find(O_3)	!visible(O_3)	+visible(O_3)
pick-up(O_3)	!grabbed(O_3) visible(O_3)	+grabbed(O_3)
go-to(O_1)	!current(O_1)	+current(O_1)

put-down(O_3, O_1)	grabbed(O_3)
	current(O_1)
		-grabbed(O_3)
		+in(O_3, O_1)

O_1:Soar Office

O_2:Copy Room

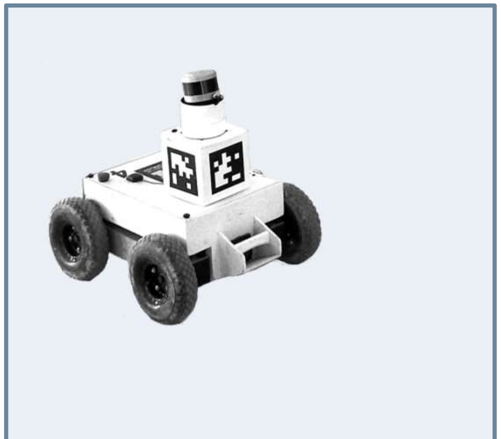
O_1: {soar, office,
current, location,
starting}
O_3: {stapler}

Task: fetch(O_3)
Goal: in(O_3, O_1)

Example: Fetch a stapler

find(O_3)

Rosie: *“I can’t find the stapler, can you help?”*



O_1:Soar Office

O_2:Copy Room

O_1: {soar, office,
current, location,
starting}
O_3: {stapler}

Task: fetch(O_3)
Goal: in(O_3, O_1)

Example: Fetch a stapler

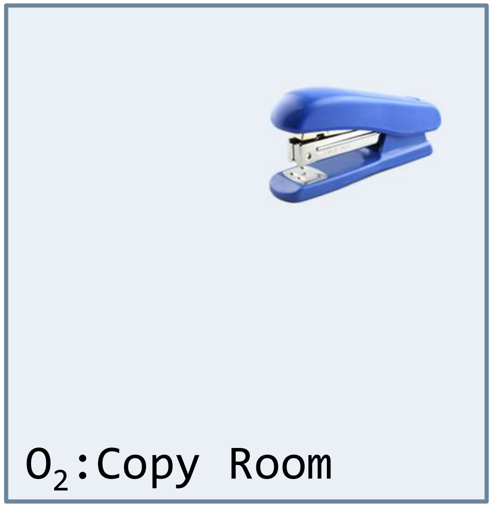
“Recall the stapler in a location”

‘the stapler’ [*uniquely id.]* -> WM O_3

‘a location’ [*type id*.] -> NEW O_5

$\texttt{recall}(O_3,\ \texttt{in}(O_5))$

O_1:Soar Office


O_2:Copy Room

O_1: {soar, office, current, location, starting}

O_3: {stapler}

Task: fetch(O_3)

Goal: in(O_3, O_1)

Example: Fetch a stapler

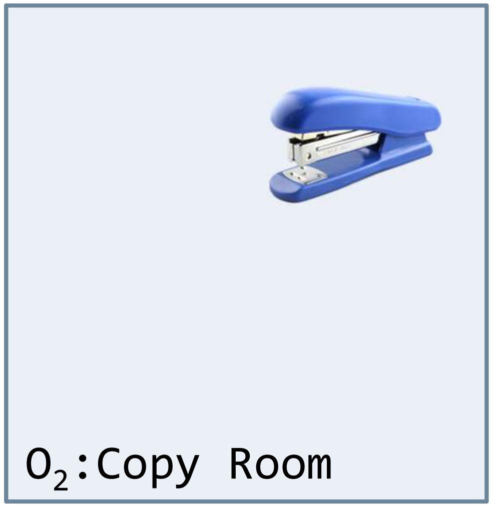
“Recall the stapler in a location”

recall(O_3, in(O_5))

Cue: { stapler(A), location(B), in(A, B) }

Result: A = O_6, B = O_2

O_1:Soar Office



O_2:Copy Room

O_1: {soar, office,
current, location,
starting}

O_2: {copy, location}

O_3: {stapler}

in(O_3, O_2)

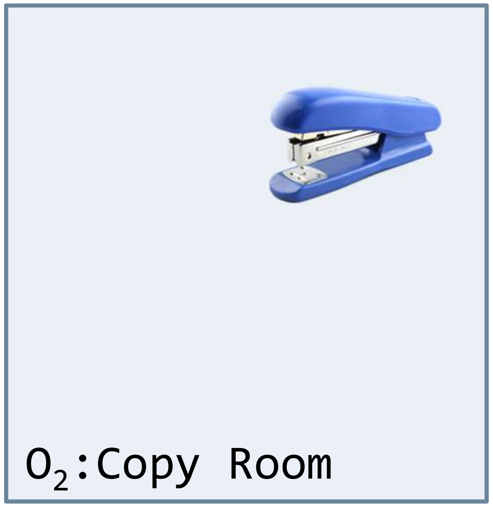
Task: fetch(O_3)

Goal: in(O_3, O_1)

Example: Fetch a stapler

go-to(O_2)

O_1:Soar Office



O_2:Copy Room

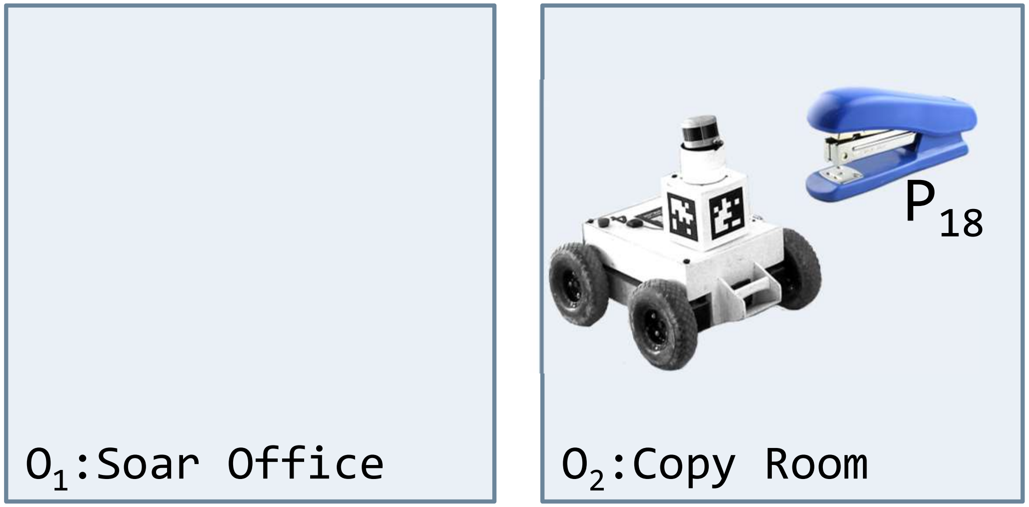
O_1: {soar, office,
current, location,
starting}
O_2: {copy, location}
O_3: {stapler}
in(O_3, O_2)

Task: fetch(O_3)
Goal: in(O_3, O_1)

Example: Fetch a stapler

go-to(O_2)

New Perception P_{18}: {stapler, blue}



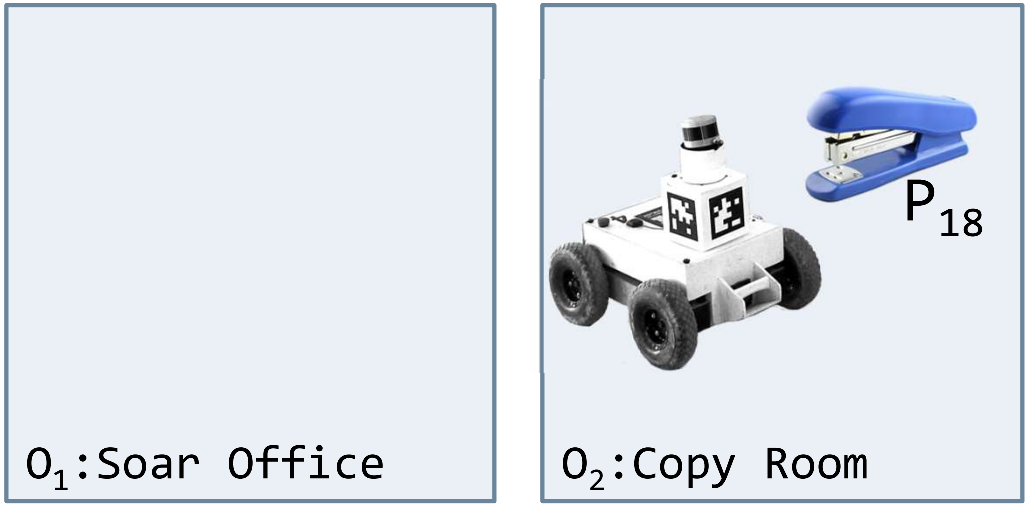
O_1: {soar, office,
location, starting}
O_2: {copy, location,
current}
O_3: {stapler}
in(O_3, O_2)

Task: fetch(O_3)
Goal: in(O_3, O_1)

Example: Fetch a stapler

go-to(O_2)

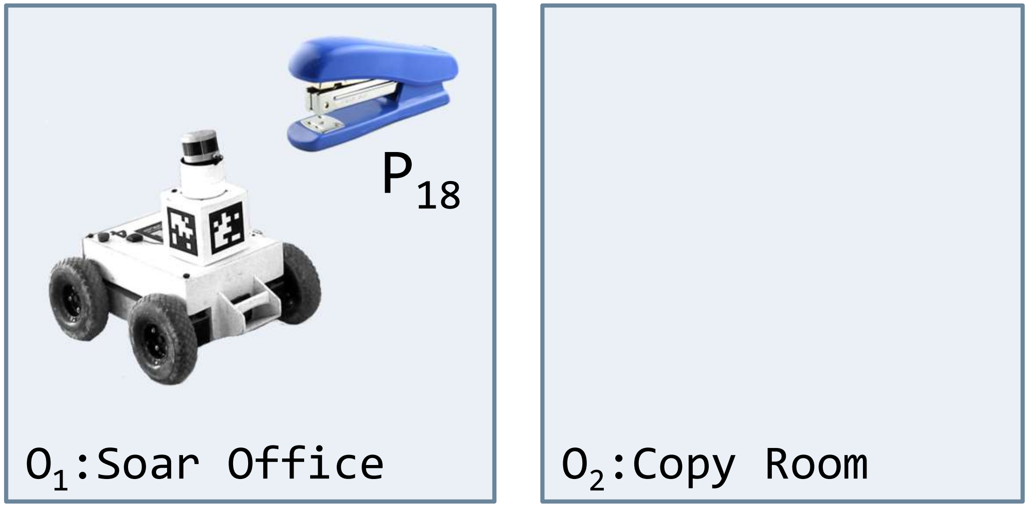
New Perception P_{18}: {stapler, blue}
Anchored to matching object O_3



O_1: {soar, office,
      location, starting}
O_2: {copy, location,
      current}
O_3: {stapler, blue,
      visible}
in(O_3, O_2)
Task: fetch(O_3)
Goal: in(O_3, O_1)

Example: Fetch a stapler

The agent completes the rest of the task and learns a policy for future instances

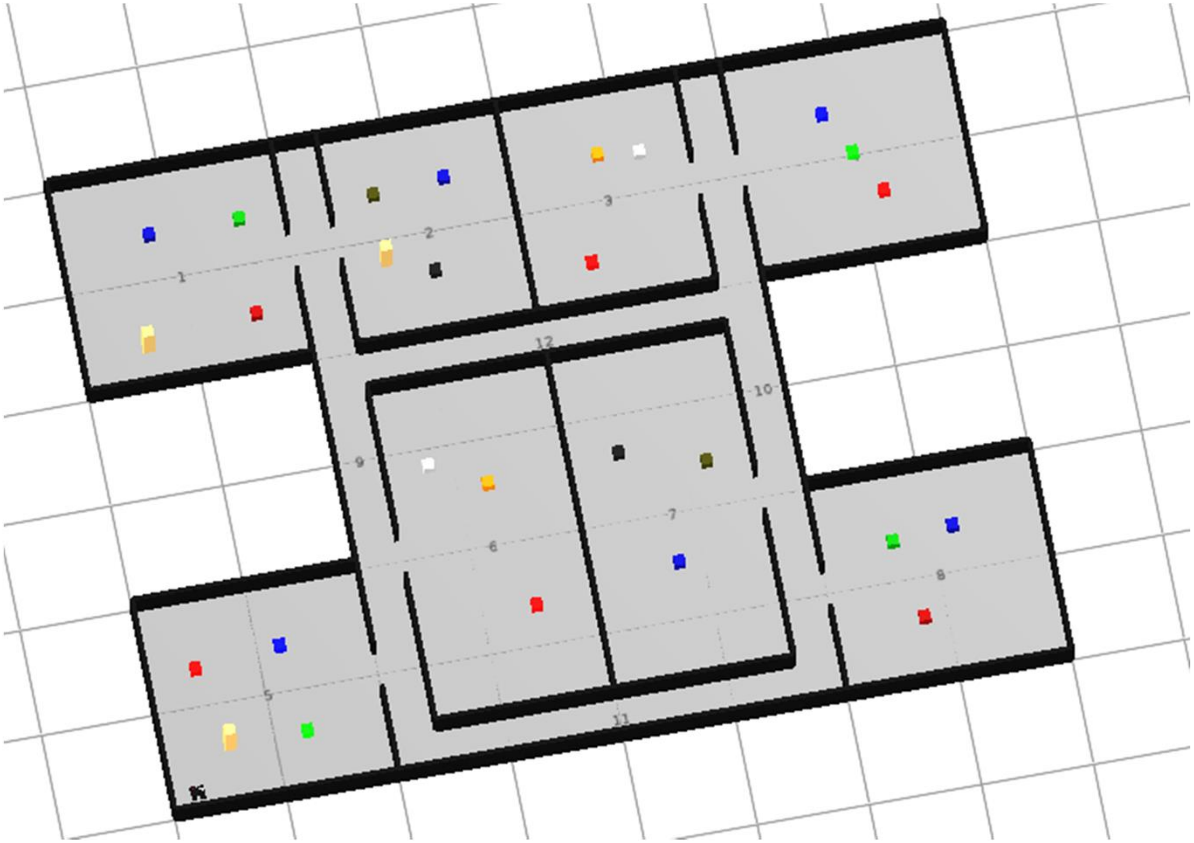


O_1: {soar, office,
 location, current,
 starting}
O_3: {stapler, blue,
 visible}
in(O_3, O_1)
Task: fetch(O_3)
Goal: in(O_3, O_1)

Task Examples

Task Command	Task Goal
Discard the package	The package is in the trash
Deliver the package to Alice	Alice is holding the package
Fetch a stapler	The stapler is in the starting location
Take the stapler to the lab	The stapler is in the lab
Tell Charlie a message	Charlie heard the message
Serve Bob	Bob is holding the desired drink
Guide Bob	You are in the desired location
Greet Alice	Alice heard the welcome message

Evaluation



- 8 rooms
- 4 hallways
- 30 objects
- 8 people

Some have default locations

Objects will move

Evaluation

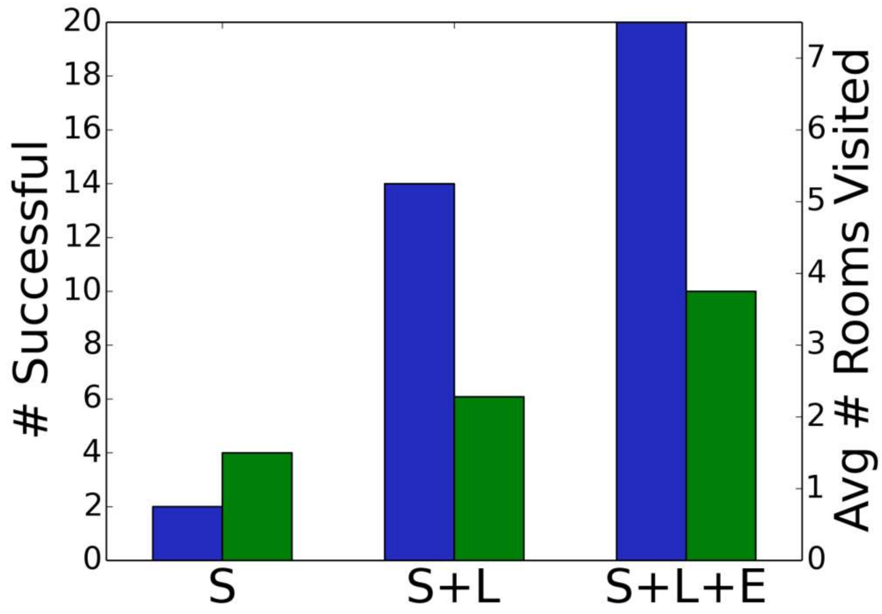
Agent asked to find 20 random objects

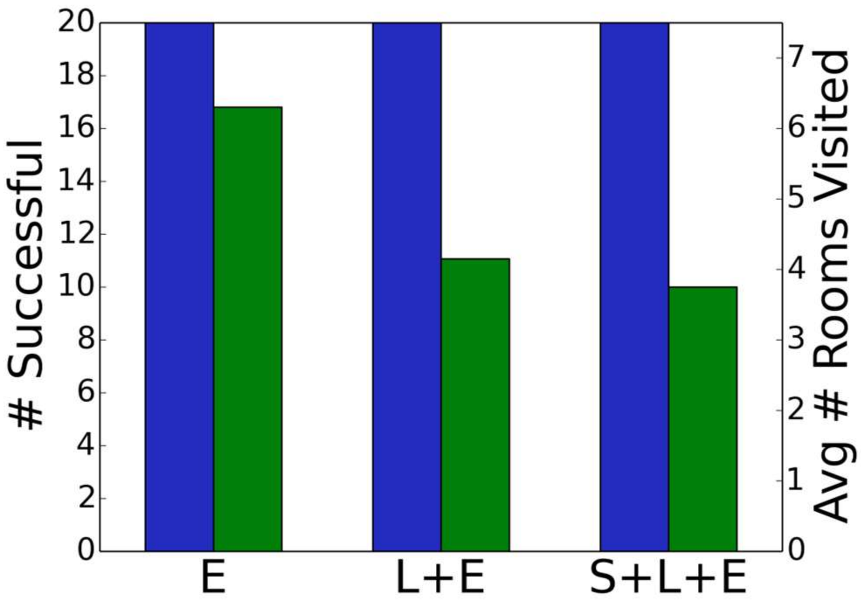
Strategies Used:

- □ Short Term Memory (S): face, go-to-location
- □ Long Term Memory (L): think, recall
- □ External Search (E): scan, explore

Measured number of successes and average number of rooms searched

Evaluation





S: Short Term Memory

L: Long Term Memory

E: External Search

Nuggets + Coal

Nuggets

- Learns tasks in a partially observable environment
- Added actions that involve the agent's memories
- Most strategies can be learned through instruction

Coal

- Limited strategies involving LTM
- Requires expertise to instruct
- Not robust to perceptual noise/errors

Questions?