

Extending Task Learning in Rosie

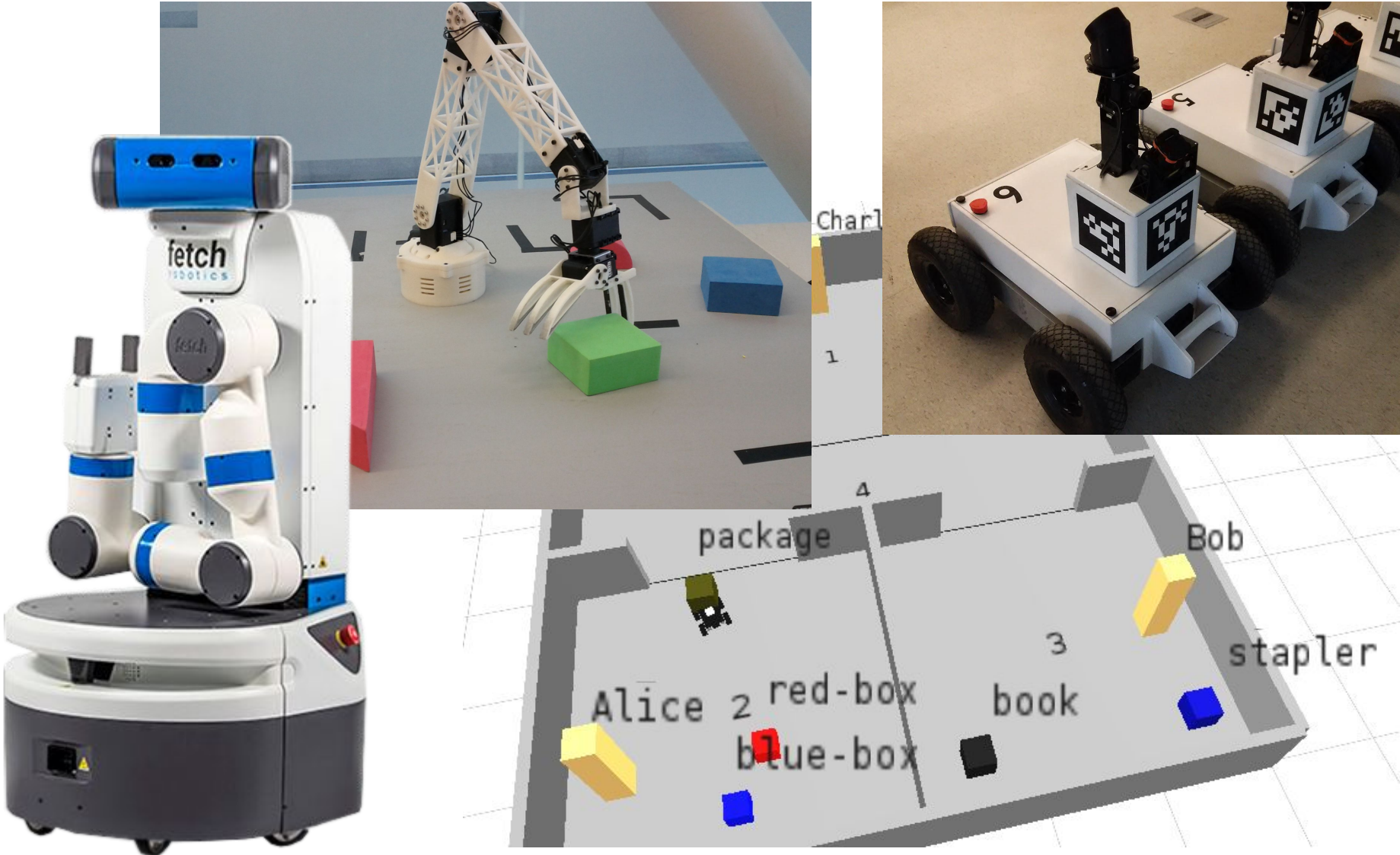
Aaron Mininger

University of Michigan

June 7, 2017

37th Soar Workshop

Task Domains



Example Tasks

Store the book on the shelf.

Deliver the package to the soar office.

Tell John a message.

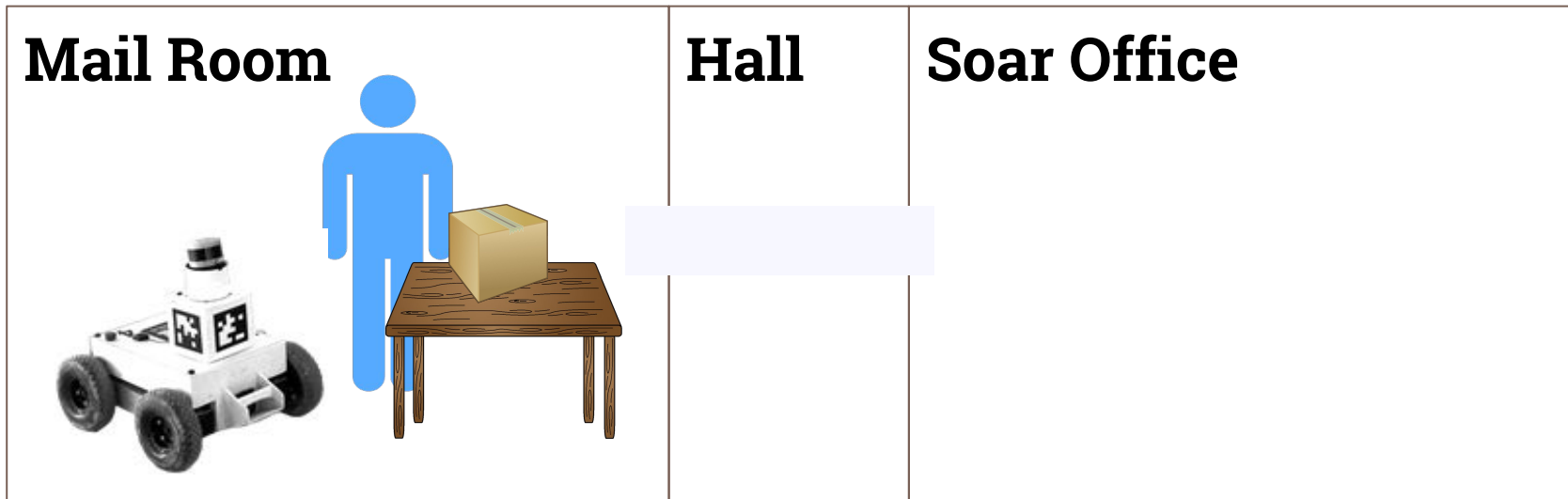
Fetch a stapler.

Serve Alice a desired drink.

Learning Deliver Task

Example Task: Deliver

Deliver the package to the soar office.



Learning Task Structure

Deliver the package to the soar office.

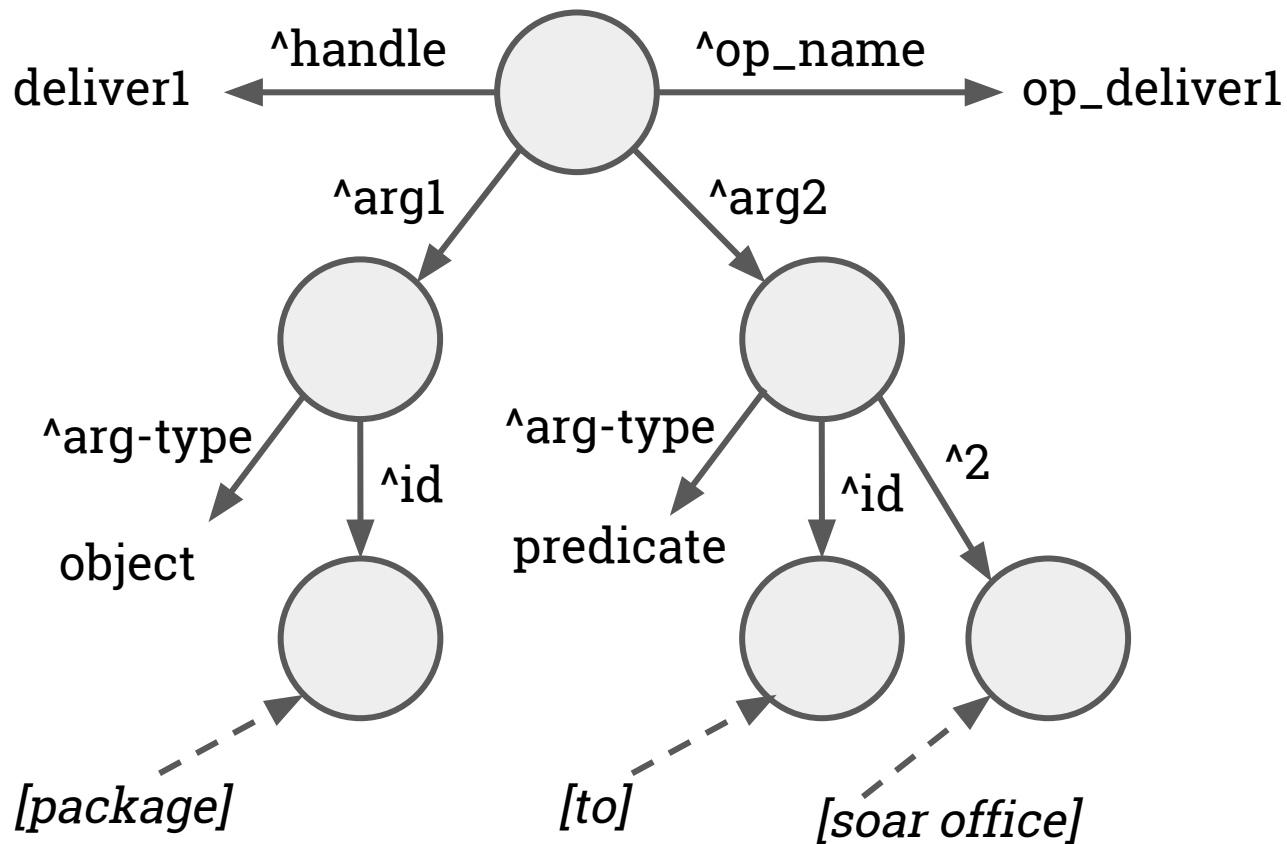
deliver has 2 arguments:

an object [package]

a predicate [to(soar office)]

Learning Task Structure

Deliver the package to the soar office.



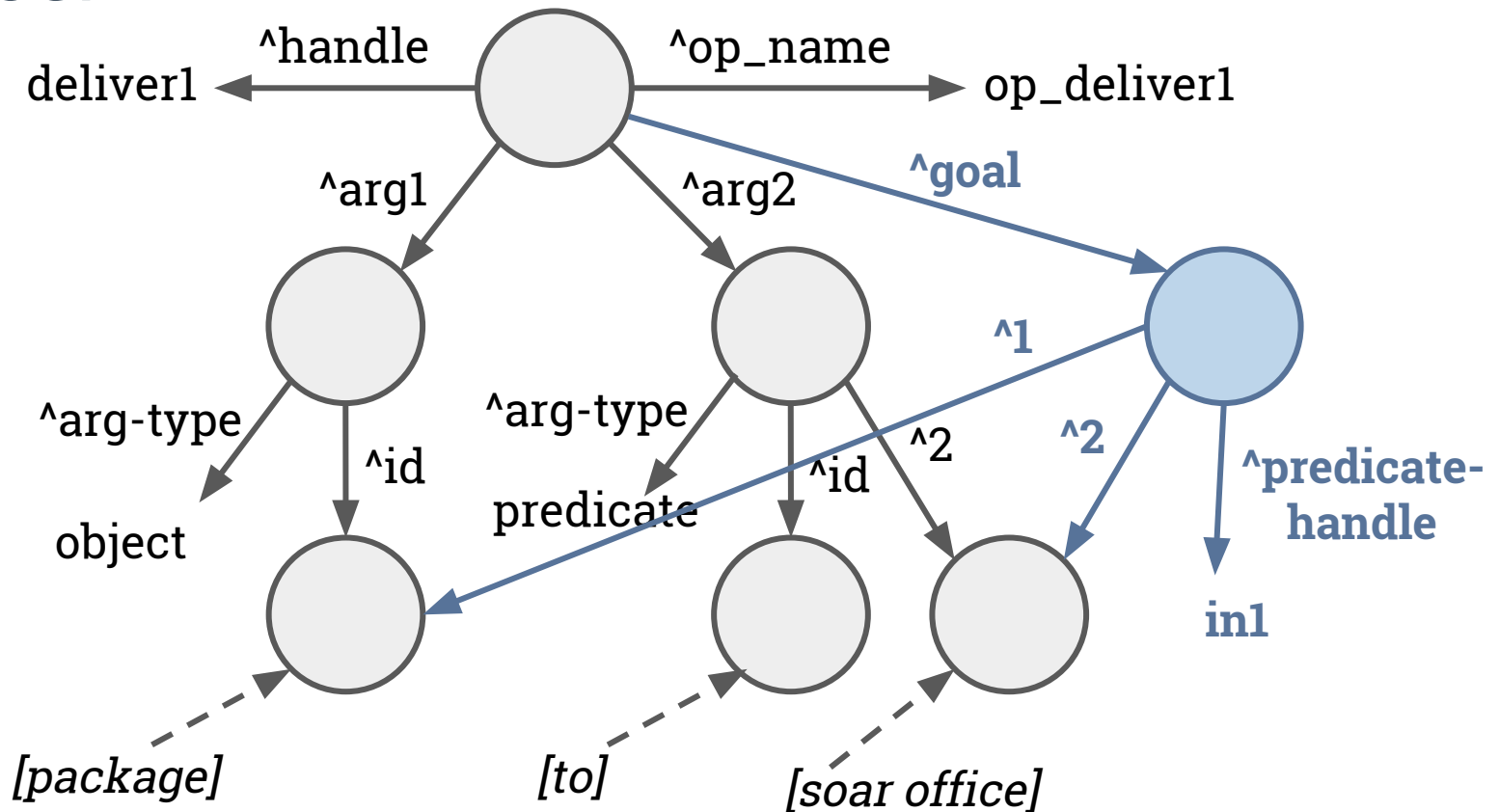
Learning Task Goal

Deliver the package to the soar office.

The goal is that the package is in the soar office.

Learning Task Goal

The goal is that the package is in the soar office.



Learning Task Decomposition

Deliver the package to the soar office.

The goal is that the package is in the soar office.

Pick up the package.

(Subtasks can come from instruction or search)

Learning Task Decomposition

Pick up the package.

Go to the soar office.

Put down the package.

We learn which sub-tasks we should consider inside the deliver problem-space

- Both a declarative and procedural representation

Learning Task Decomposition

Pick up the package.

```
sp {deliver*propose*pick-up
  (state <s> ^name execute-task
    ^task <t>)
  (<t> ^handle deliver1 ^arg1.obj <obj>)
-->
  (<s> ^operator <o> +)
  (<o> ^name op_pick-up1
    ^arg1 <obj>)
}
```

Learning Task Policy

Deliver the package to the soar office.

The goal is that the package is in the soar office.

Pick up the package.

Go to the soar office.

Put down the package.

Learning Task Policy

- We now have an example execution, we want to learn a general policy
 - **State-based policy**
 - Rule that says in state S , do subtask T
 - *if delivering X to Y , holding X , and not in room Y , then drive to room Y*
 - **Flexible**
 - Not just memorizing a procedure
 - **Learn from one example**

Explanation Based Learning

- Insight: "It is possible to form a justified generalization of a single example if a system has explanatory capabilities"*
 - If the agent can explain why the actions it took led to the goal, then it can form a justified generalized policy
 - We need to 'replay' the actions

*Explanation-Based Generalization: A Unifying View (Mitchell and Keller, 1986)

Retrospective Learning

Retrieve the initial state and the subtasks from episodic memory

Internally apply the subtasks and simulate the effects

Once the goal is achieved, we learn a new preference rule through chunking

Retrospective Learning

```
sp {deliver*prefer*pick-up
  (state <s> ^name execute-task
    ^task <t> ^world <w>
    ^operator <o> +)
  (<t> ^handle deliver1 ^arg1.obj <obj> ^arg2.obj <loc>)
  (<w> ^current-location { <cur-loc> <> <loc> }
    -^holding <any>)
  (<obj> ^visible true)
  (<o> ^name op_pick-up1 ^arg1.obj <obj>)
-->
  (<s> ^operator <o> >)
}
```

Task Learning Recap

- Rosie learns the following:
 - **Task Structure**
 - **Task Goal**
 - **Task Decomposition**
 - **Task Policy**

Restrictions and Limitations

- The kinds of tasks Rosie can learn are fairly restrictive:
 - Subtasks must be single, discrete actions
 - Goal must be a conjunction of known predicates
 - Action models must be sufficient to achieve the goal

Future Extensions

Future Extensions

- Three Main Areas:
 - **Control Structures**
 - for each, while, until, if
 - **Procedural Tasks**
 - give a tour, patrol a route
 - **Incomplete Action Models**
 - use a dishwasher

Object Groups

- Handle multiple objects in a goal/task
 - **Numerical Quantities**
 - Deliver 2 packages to the soar office.
 - The goal is that there are at least 4 sodas in the fridge.
 - **Universal Quantifiers**
 - The goal is that all cans are in the trash.
 - Stack all the blocks on the table.
 - **Enumeration**
 - Store each object on the table.
 - Maintain each location on this floor.

Conditional Statement

- **Conditionally execute a subtask**
 - **If Statement**
 - Example: teaching a maintenance task
 - If you are in the kitchen, stock the fridge.
 - If you are in an empty conference room, turn the lights off
 - **Whenever Statement**
 - Independent task performed when appropriate
 - Whenever you see trash, throw it away

Looping Statements

- Execute a subtask multiple times
 - **Until Clause**
 - The condition can be used as the action model
 - Wait until *the steak is cooked*.
 - **While Clause**
 - Defines a 'negative' goal, which is harder to learn
 - While the table is not clear, store an object on the table.
 - Wait while the steak is uncooked.

Procedural Tasks

- Goal is not to achieve some desired state
- but to execute a procedure
 - Patrol a route
 - Give a tour
 - Perform a dance routine
- **Learn the desired sequence of tasks**
 - Should include the previous control structures

Compositional Tasks

- Goal is to execute a set of subtasks
- ordering doesn't matter
 - Maintenance task - do a list of tasks
 - Can specify priorities like *first, last, before, after*

Incomplete Action Models

- What if there is a goal, but our models are insufficient to model its satisfaction
 - Mix of procedural and goal-oriented tasks
- **Example: Fill a cup with water**
 - The goal is that the cup is full of water
 - Put the cup under the faucet
 - Turn on the faucet
 - Wait until the cup is full of water
 - Turn off the faucet

Example Tasks

Maintenance Task

Do a maintenance task.

Tidy up each location.

Go to the location

If you are in the kitchen, stock the fridge

The goal is that there are at least 8 sodas in the fridge

If there are not at least 8 sodas in the fridge, move a soda from the shelf into the fridge

If you are in an empty conference room and the lights are on, turn the lights off

If you are in the library and it is empty, store all the books on the table.

Prepare a Room

Prepare the conference room for a presentation.

Do the following tasks:

First, if the lights are off turn the lights on.

Put 4 waters on the back table.

Put 4 sodas on the back table.

Lower the projector screen.

Finally, if the speaker is not in the room then find the speaker and lead them to the room.

Nuggets and Coal

Nuggets and Coal

Nuggets

- Greatly expand the breadth and complexity of tasks we can teach
- Flexibly combine different types of tasks hierarchically

Coal

- Not yet implemented
- Difficult to get all the pieces working together
- Likely require fairly restrictive language usage

Questions?