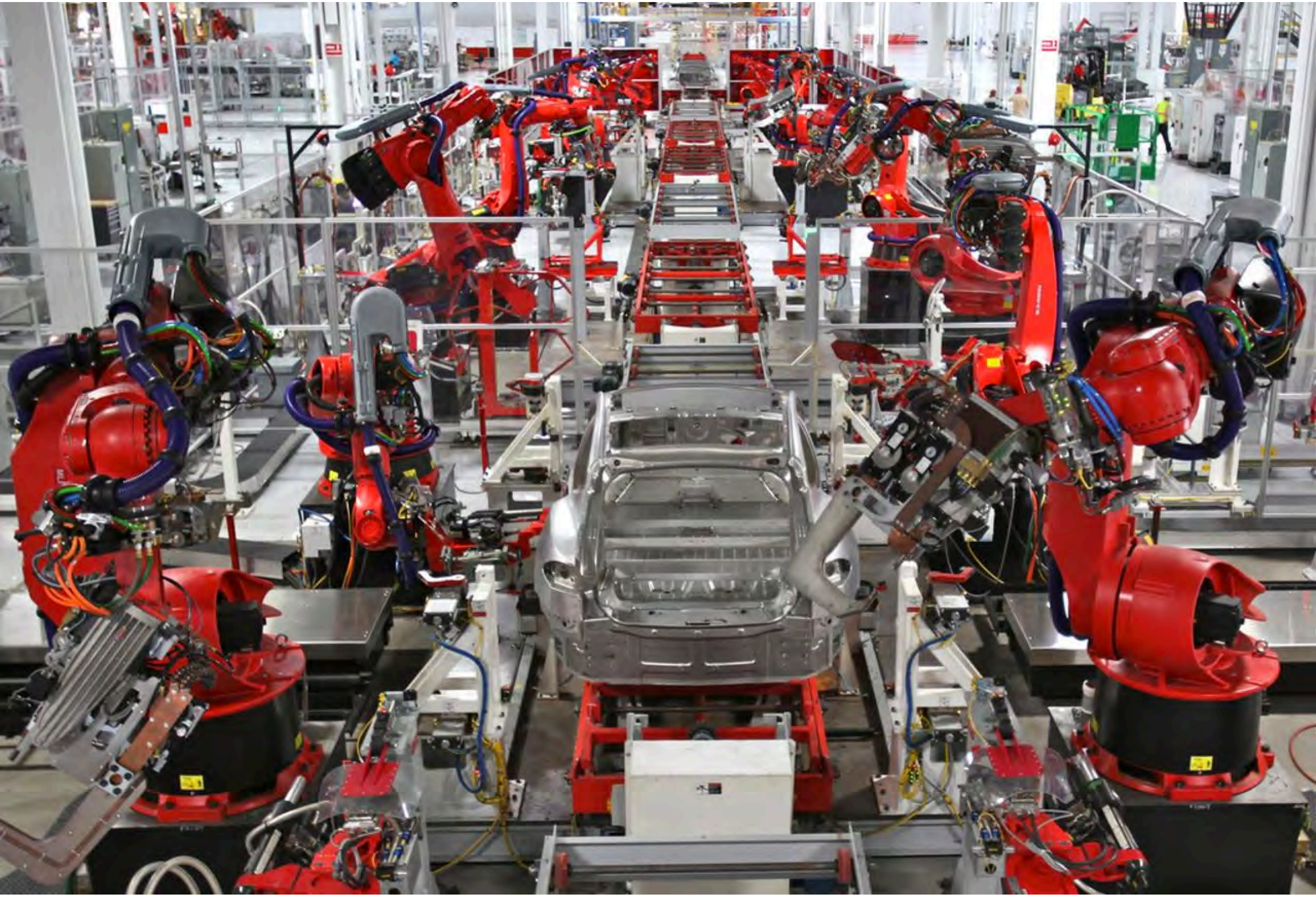
Hierarchical Learning for Human-Robot Collaboration

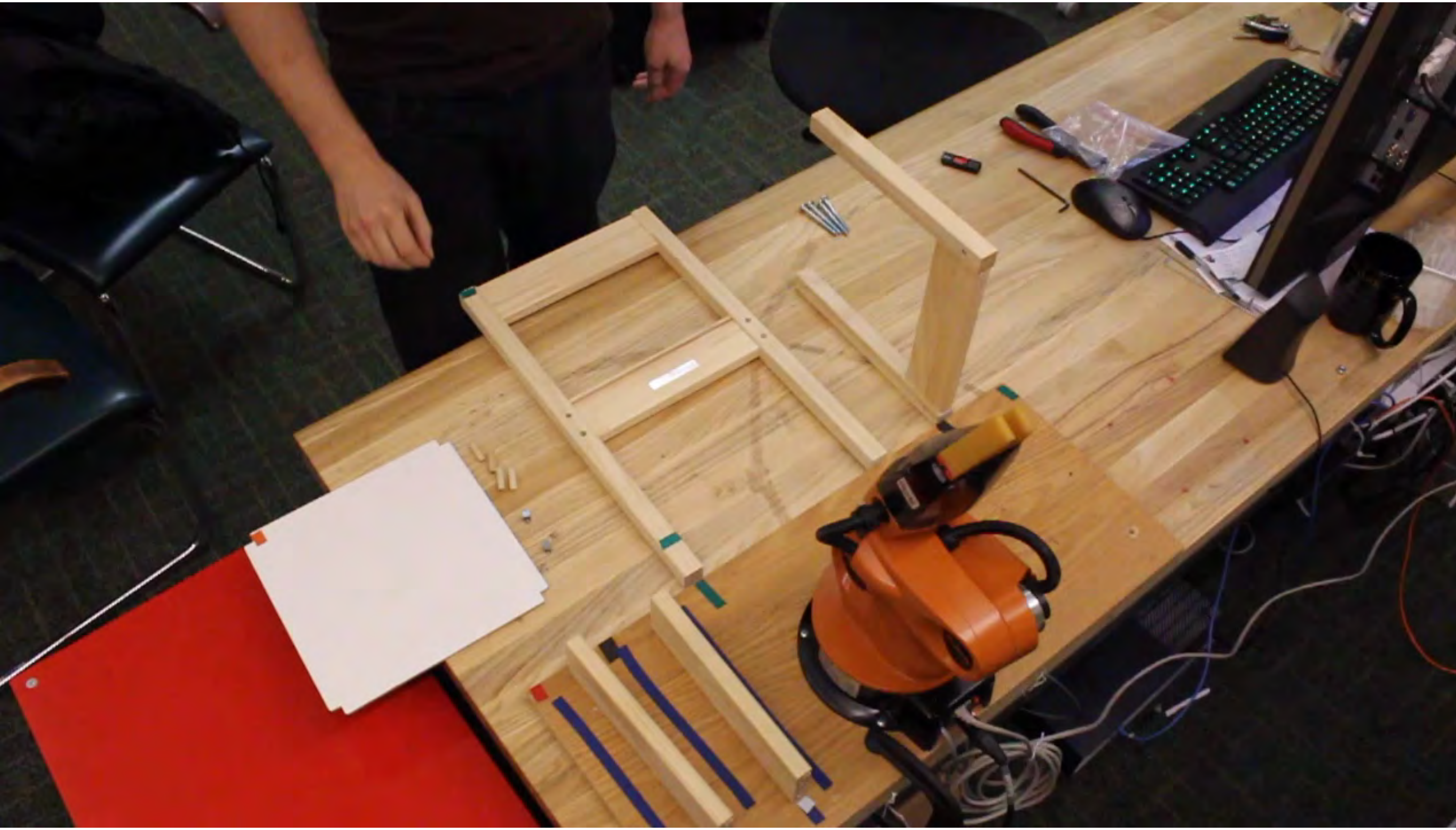
Brian Scassellati

scaz@cs.yale.edu

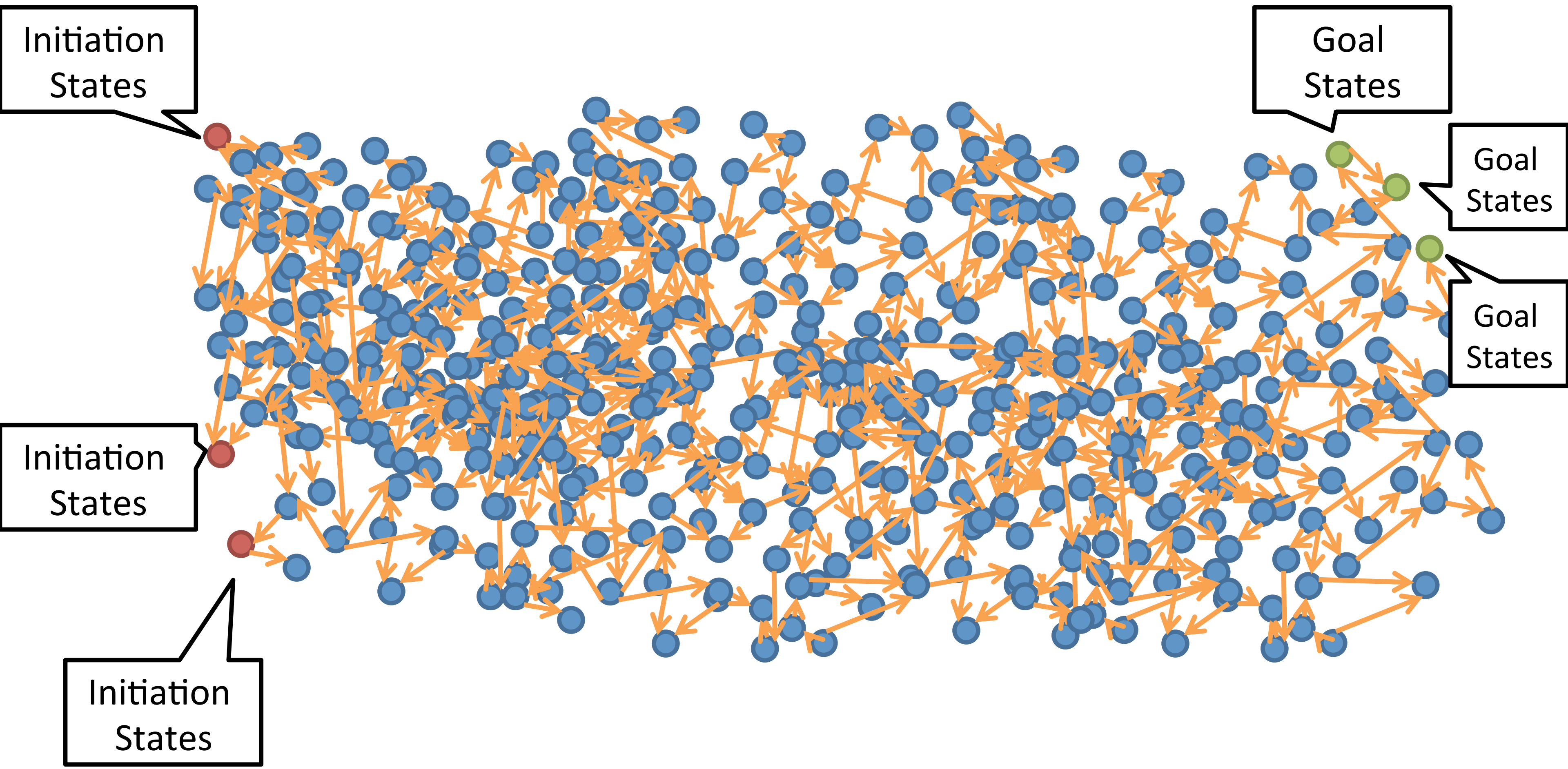
http://scazlab.yale.edu



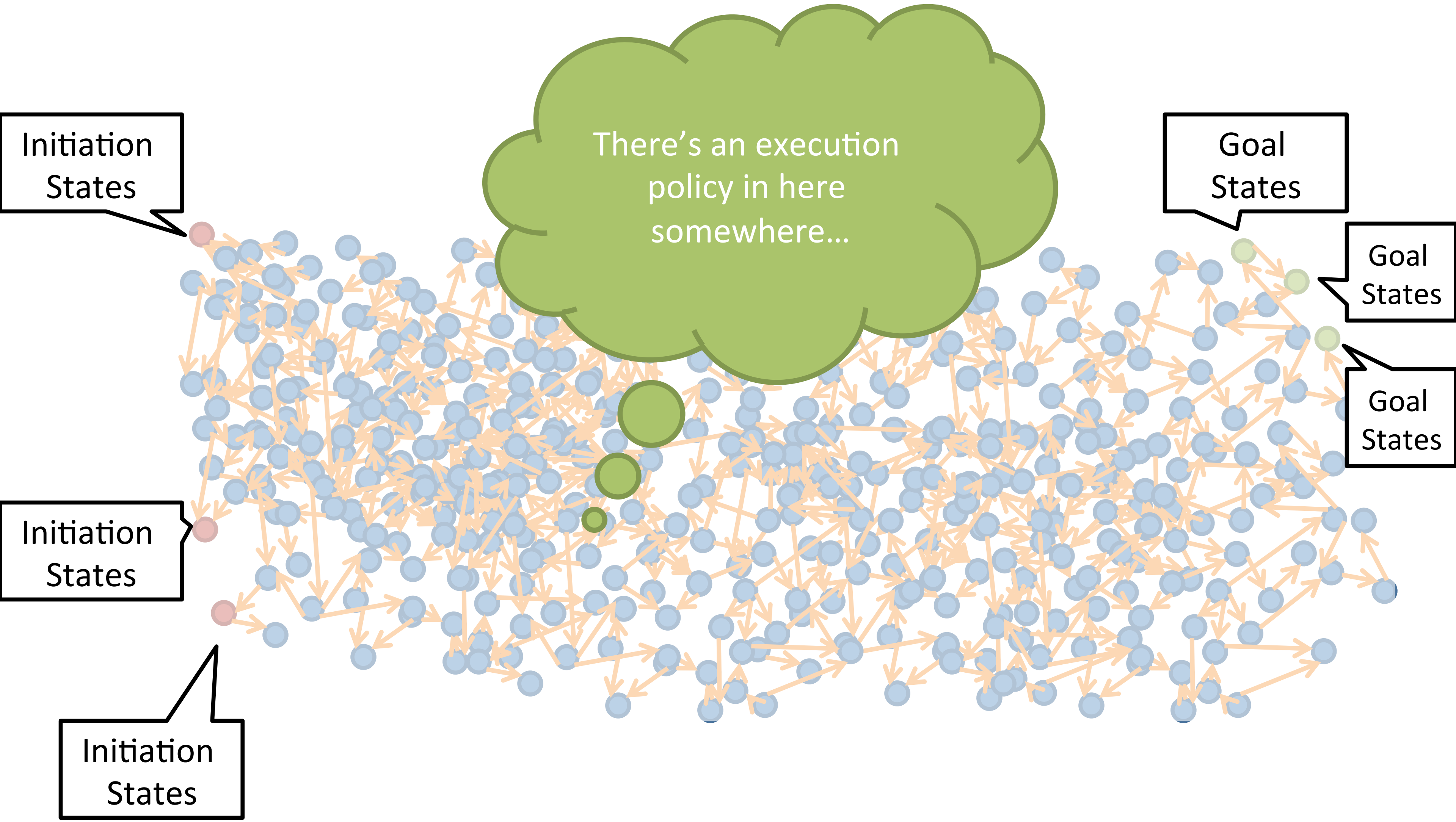
Sequential Manipulation Tasks



SMDPs Modeling Tasks Are Complex



SMDPs Modeling Tasks Are Complex



Near-Term Assistive Scenarios

Changes as we consider Collaborative Manufacturing

- Keep many of our pillars
  - Observations of sequential manipulation tasks
  - Existing methods for learning from demonstration
  - Focus on execution policies
- Adapt to collaborative setting
  - Move away from flat representations
  - Move away from divide-and-conquer planning mechanisms
  - Consider collaboration in a broad sense

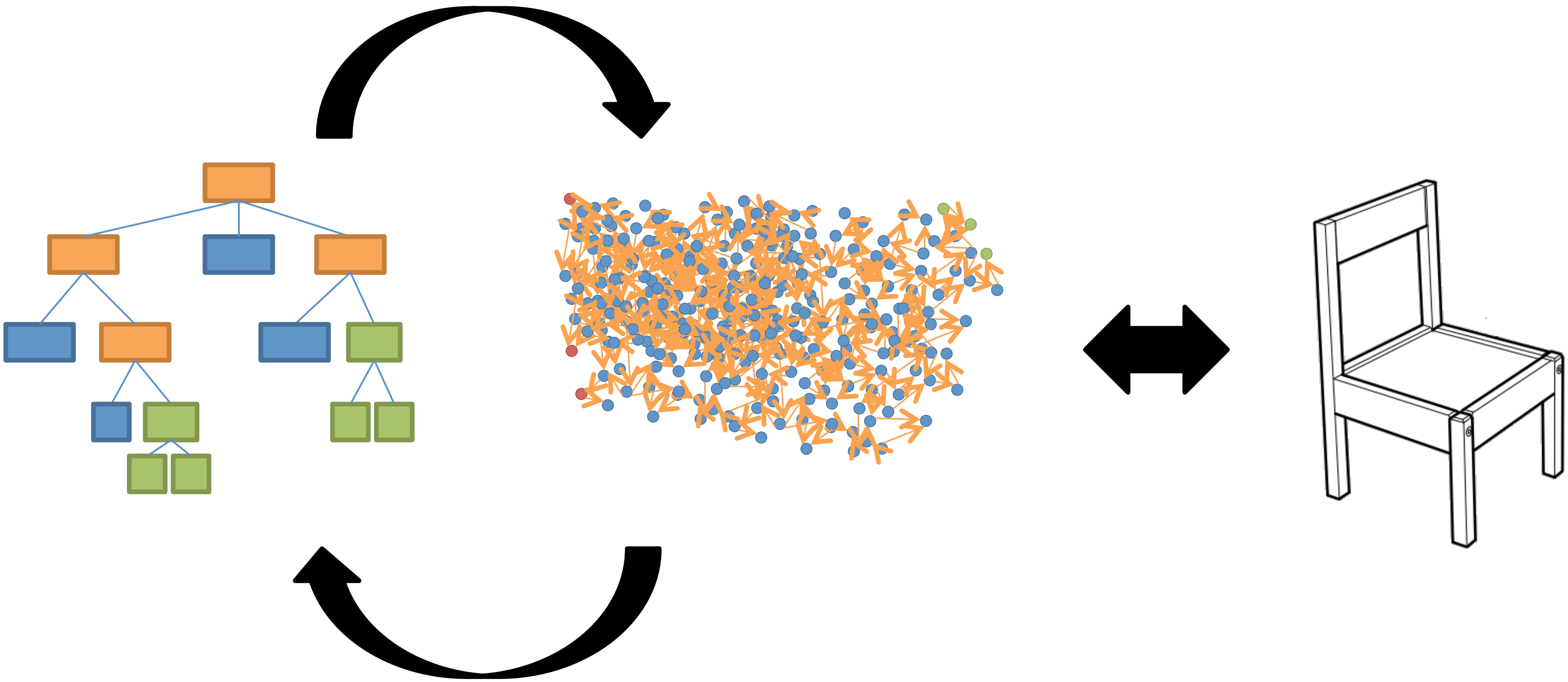
Changes as we consider Collaborative Manufacturing

- Keep many of our pillars
  - Observations of sequential manipulation tasks
  - Existing methods for learning from demonstration
  - Focus on execution policies
- Adapt to collaborative setting
  - Move away from flat representations
  - Move away from divide-and-conquer planning mechanisms
  - Consider collaboration in a broad sense

Benefits of Hierarchical Structure

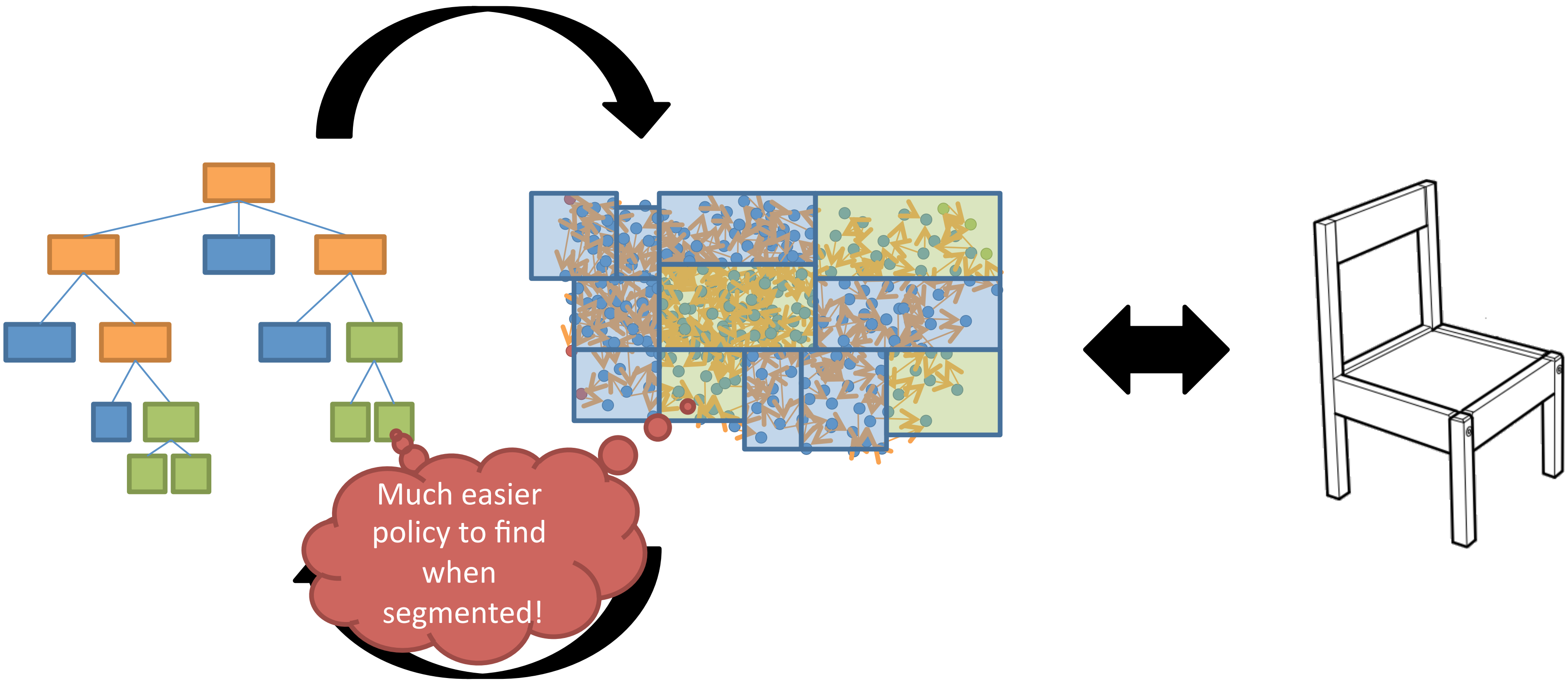
- Benefits of hierarchical structure:
  - Reduce dimensionality of policy search space

Leverage Hierarchical Structure



Hierarchical structure within the SMDP can be used to make the policy search more tractable

Leverage Hierarchical Structure

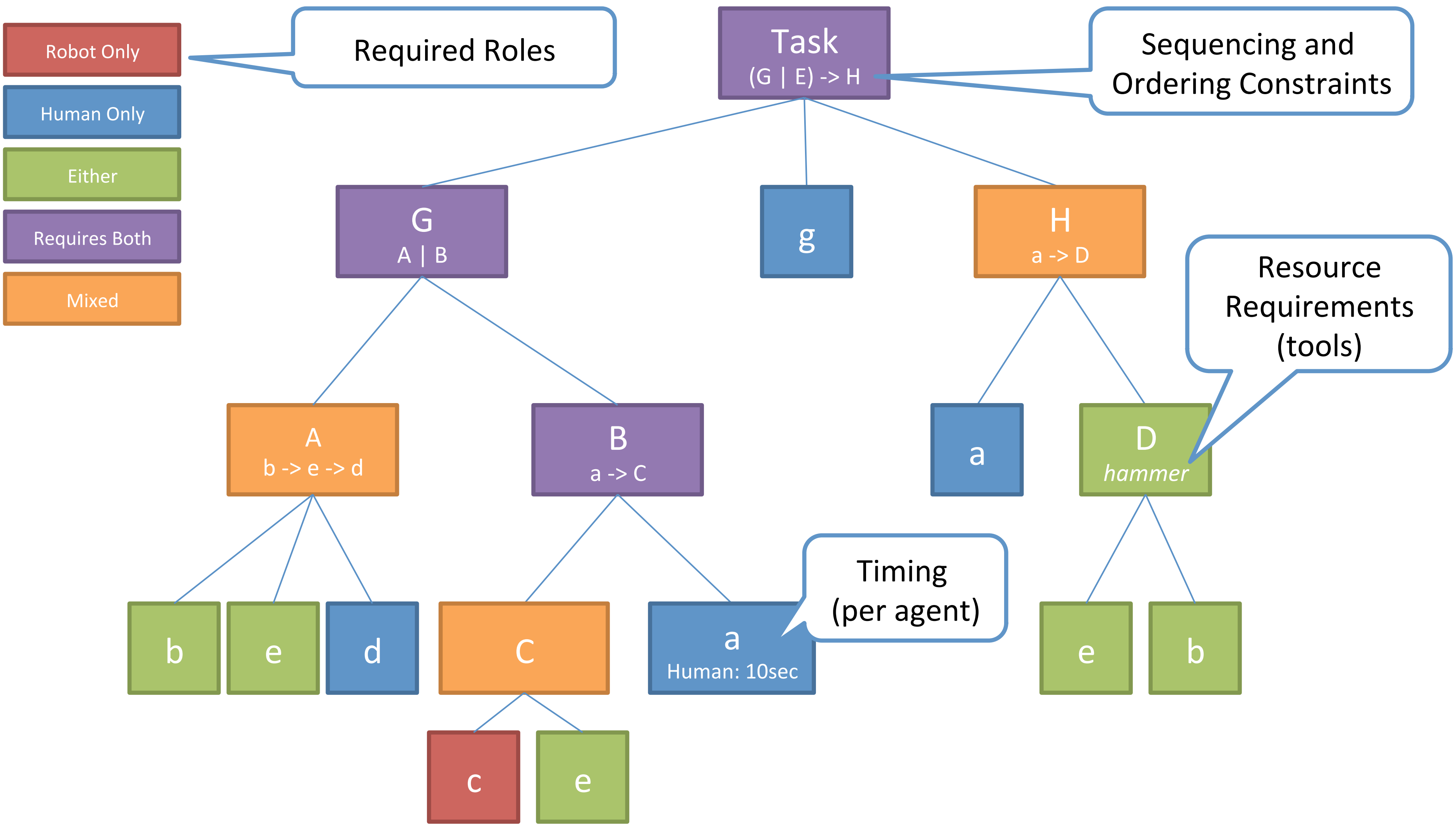


Hierarchical structure within the SMDP can be used to make the policy search more tractable

Benefits of Hierarchical Structure

- Benefits of hierarchical structure:
  - Reduce dimensionality of policy search space
  - Multi-agent task allocation
    - Parallel execution
    - Preferential allocation

Augmenting Hierarchical Plans with Social Metadata



Benefits of Hierarchical Structure

- Benefits of hierarchical structure:
  - Reduce dimensionality of policy search space
  - Multi-agent task allocation
    - Parallel execution
    - Preferential allocation
  - Transparency
    - Similarity of cognitive models
    - Ability to leverage communication

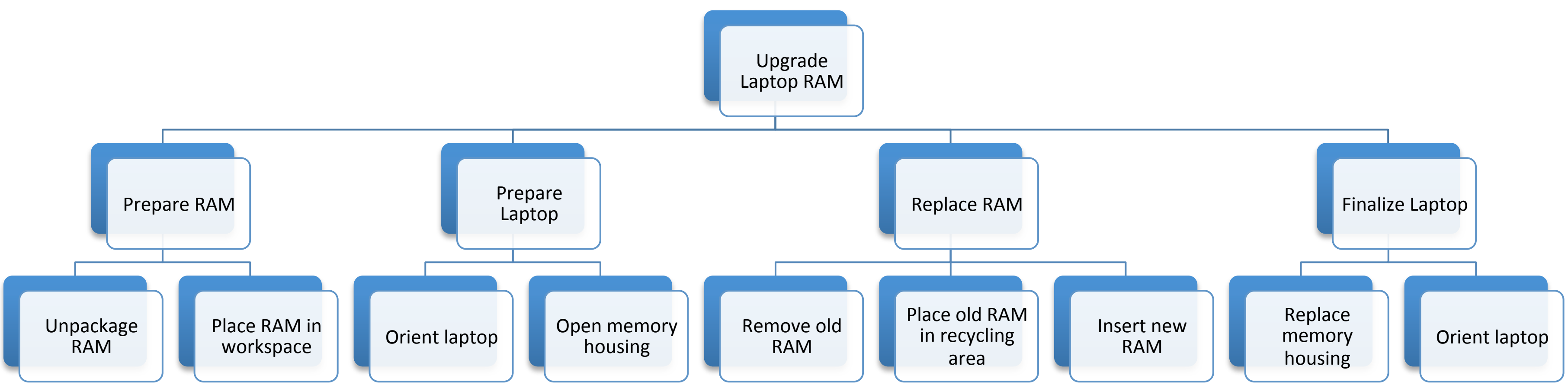
Benefits of Hierarchical Structure

- Benefits of hierarchical structure:
  - Reduce dimensionality of policy search space
  - Multi-agent task allocation
    - Parallel execution
    - Preferential allocation
  - Transparency
    - Similarity of cognitive models
    - Ability to leverage communication
- Disadvantages:
  - How do we build the hierarchy from observation?

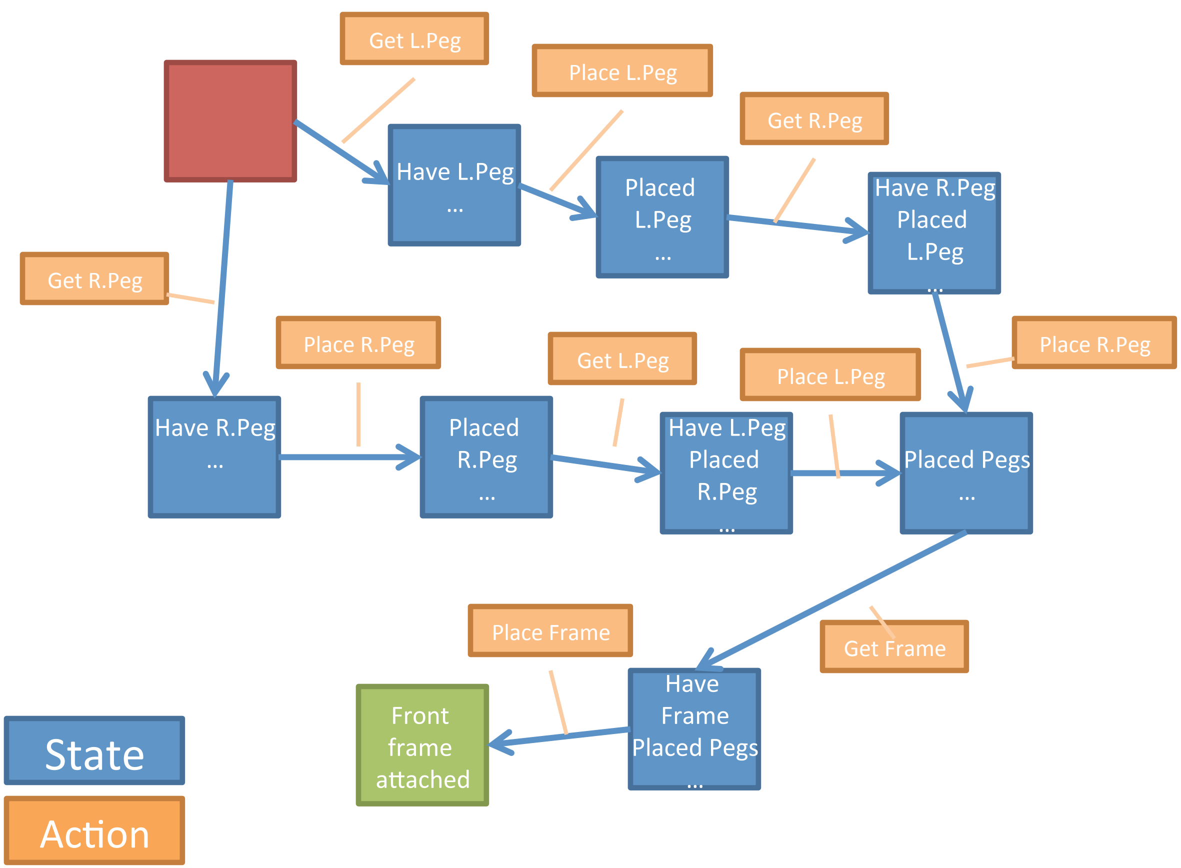
Building Hierarchical Structure from Sequential Observations

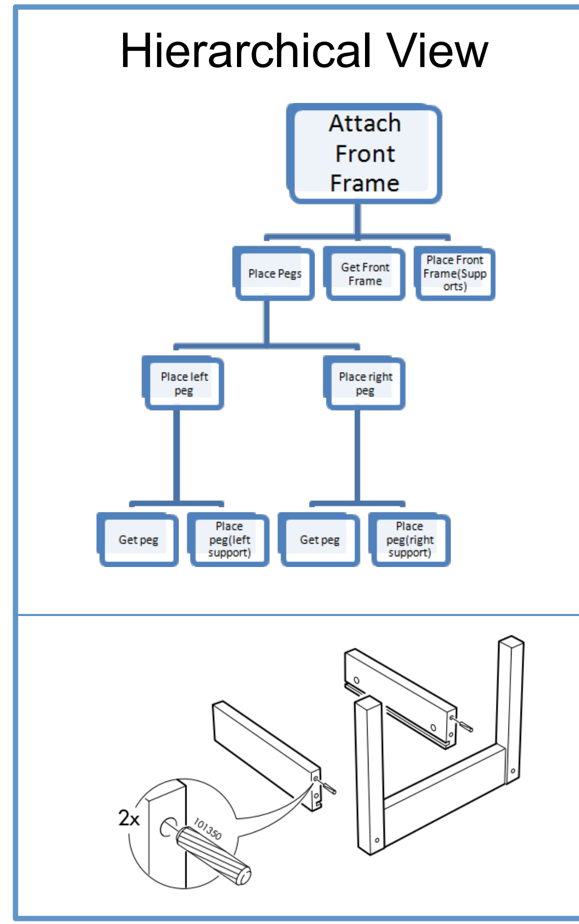
Performance by worker "A"

Performance by worker "B"



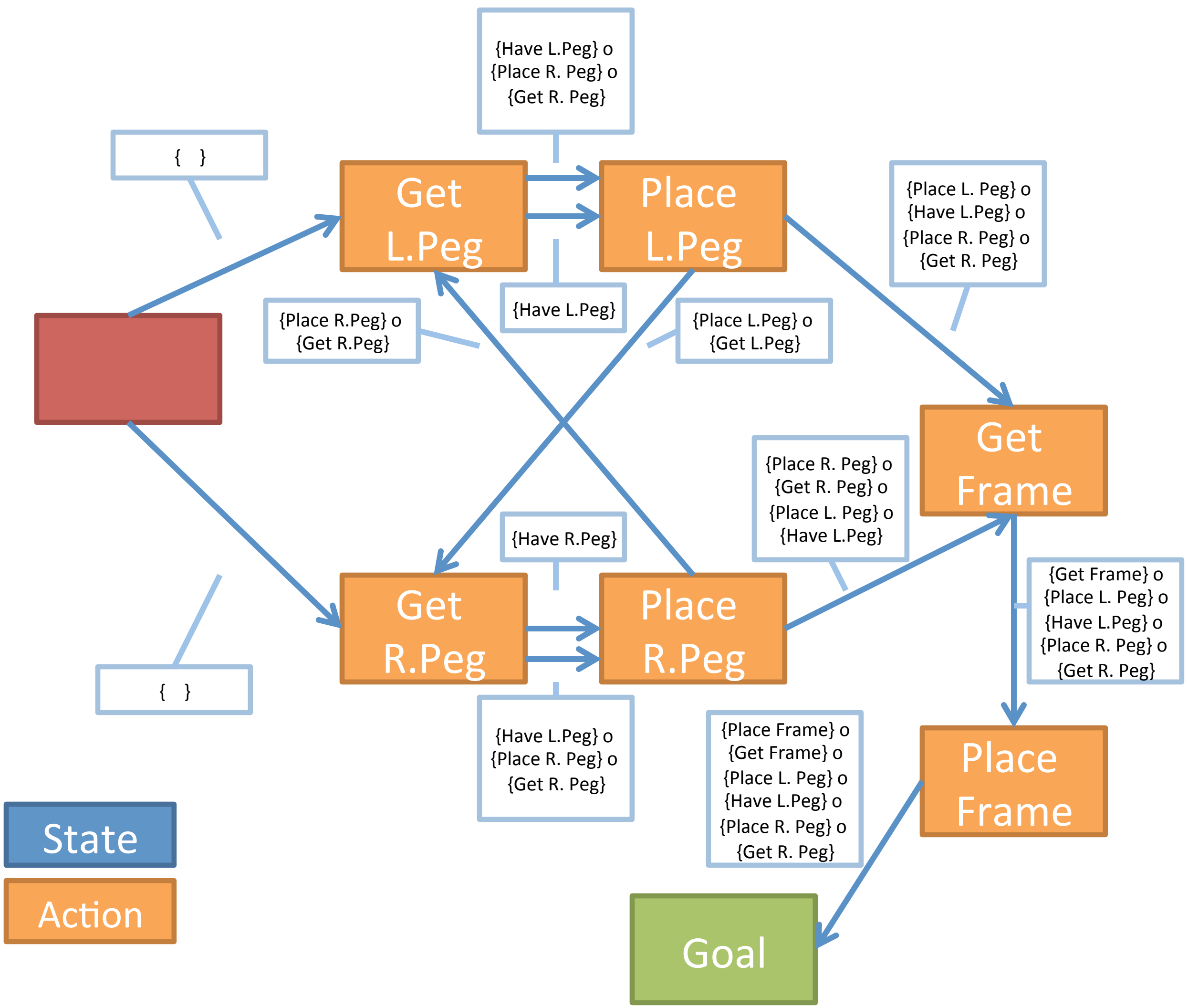
SMDP of “Attach Front Frame” Subtask





(Hayes & Scassellati, IROS 2014)

SMDP-Conjugate of “Attach Front Frame” Subtask

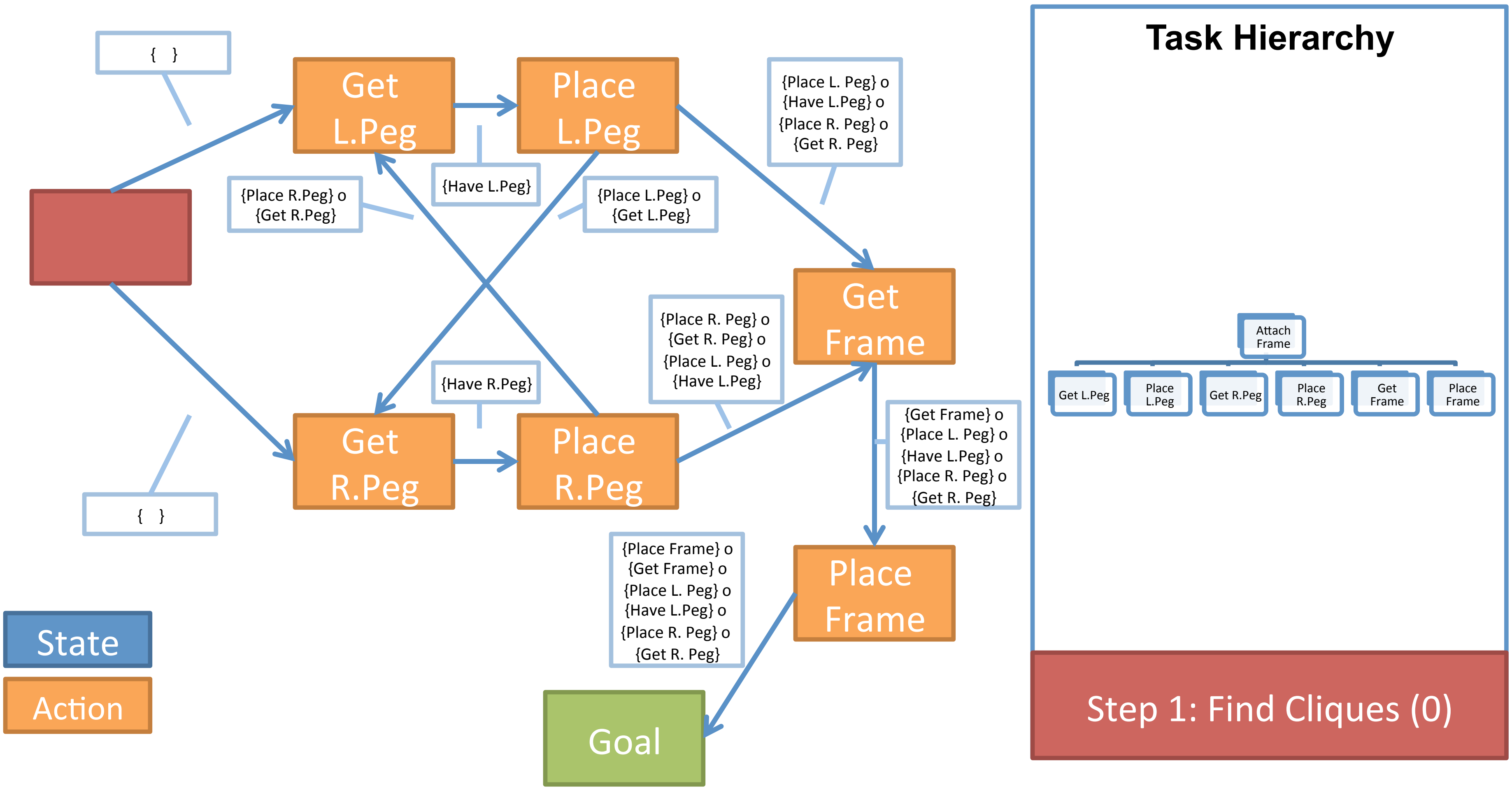


SMDP Conjugate: Actions become vertices and required state is described on edges as a composition of motor primitives.

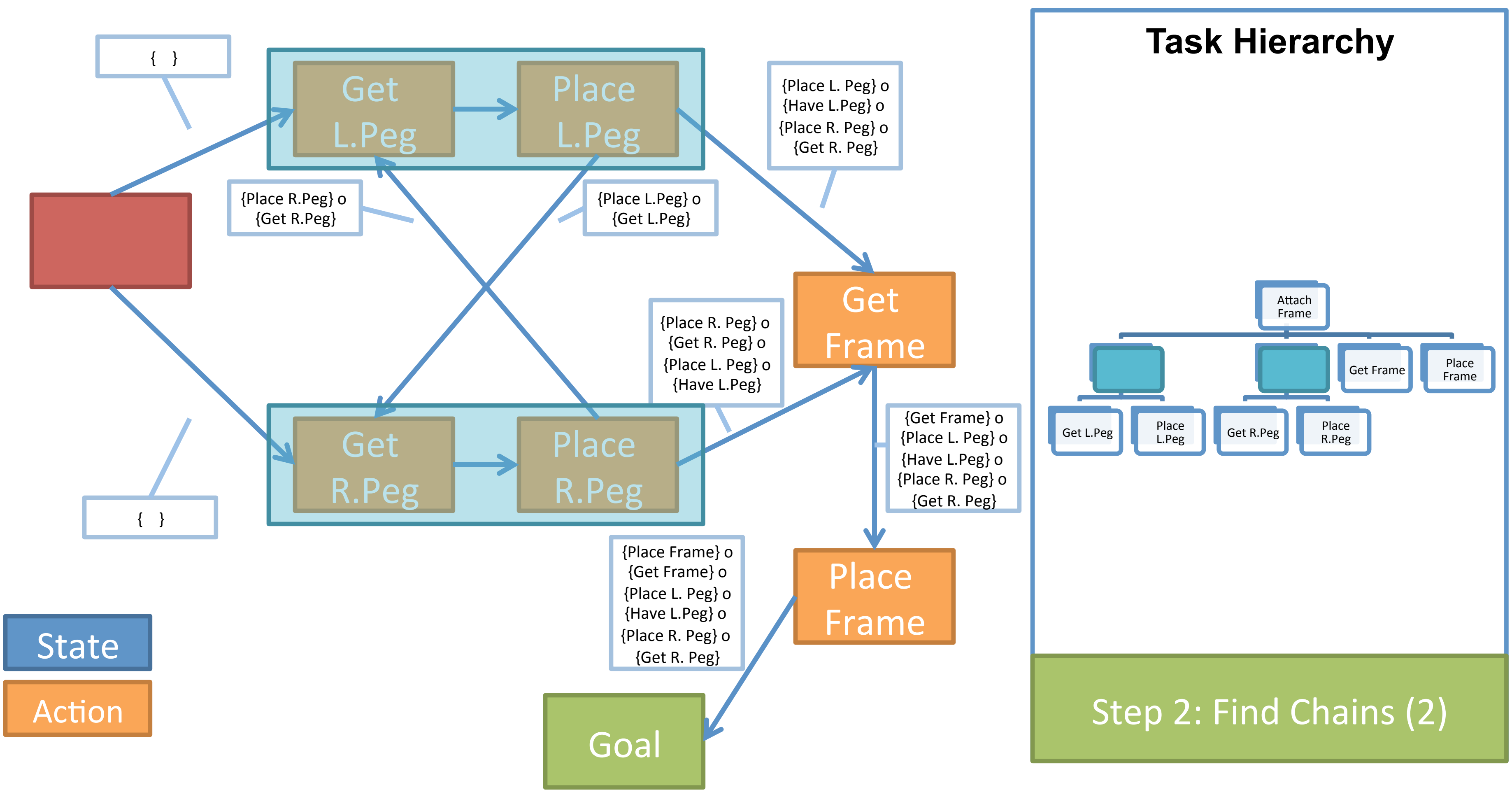
Edges are labeled with transition **requirements** – A composition of motor primitives describing the world state required to use that edge.

Vertices contain motor primitives that can be executed only upon **arriving** in the node.

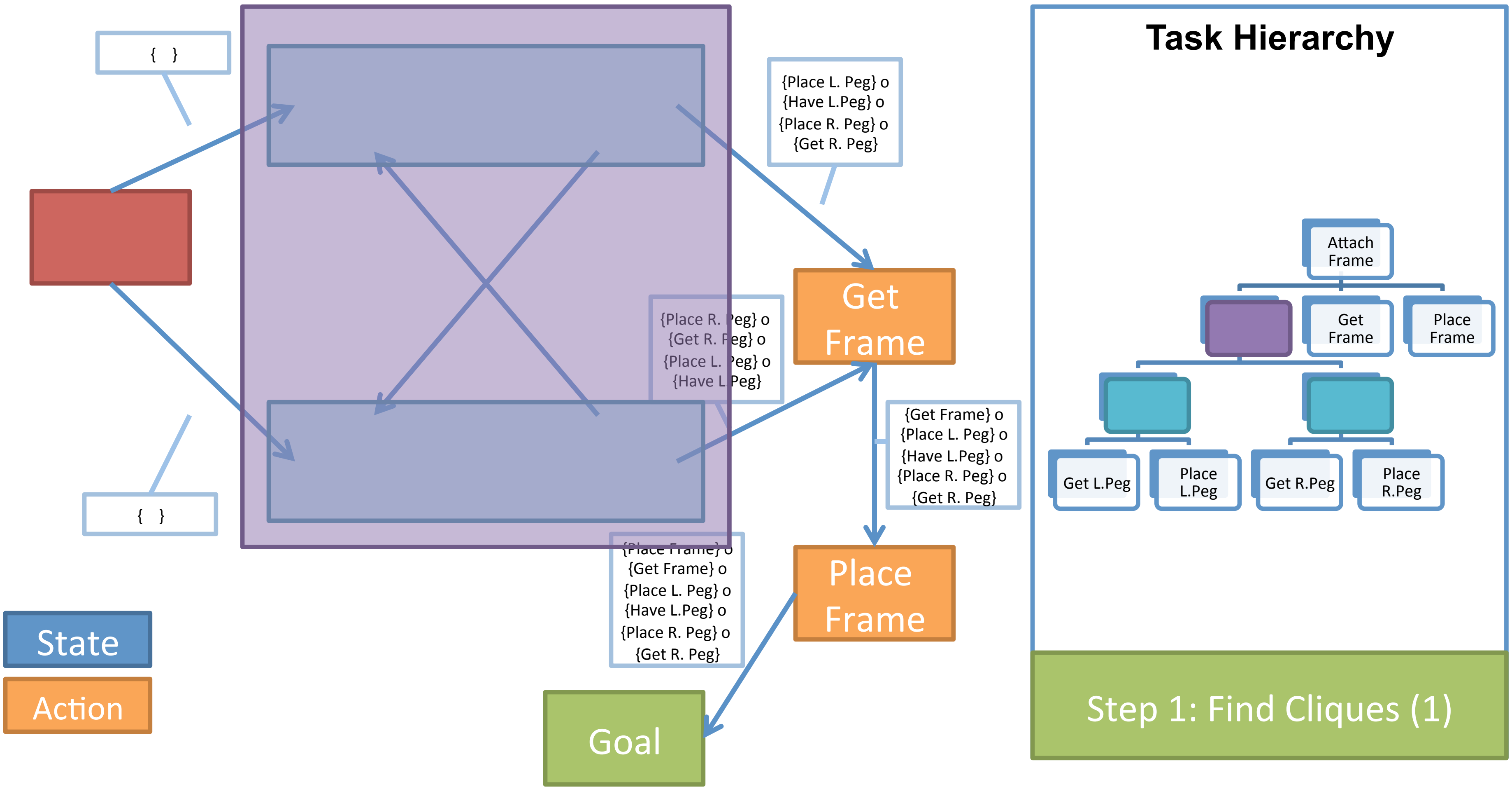
Building Hierarchical Structure



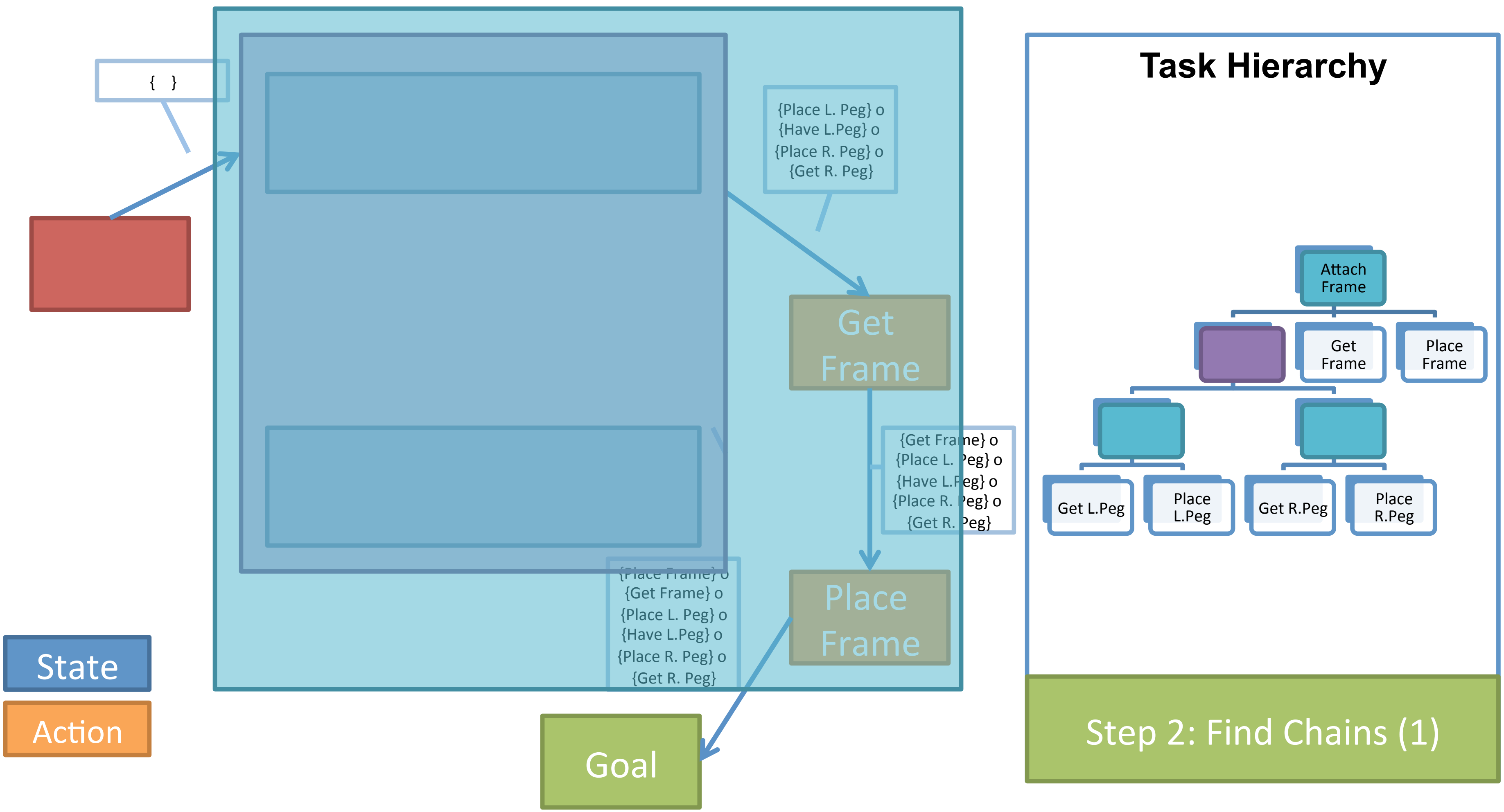
Building Hierarchical Structure



Building Hierarchical Structure



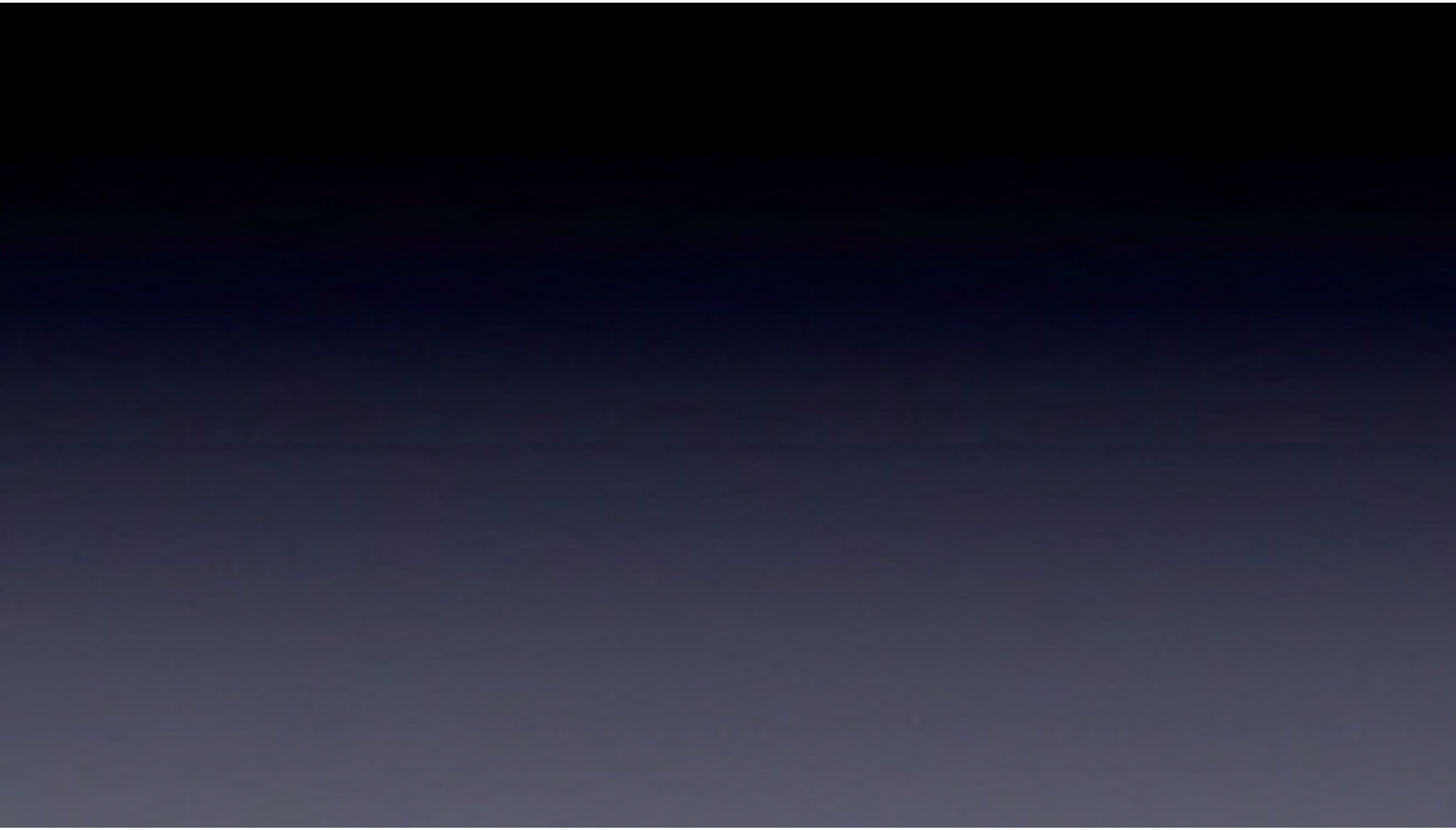
Building Hierarchical Structure



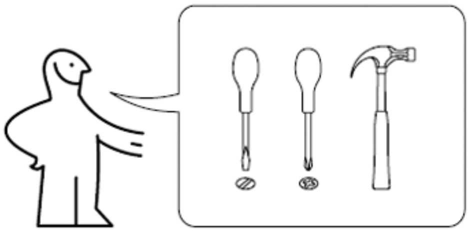
Changes as we consider Collaborative Manufacturing

- Keep many of our pillars
  - Observations of sequential manipulation tasks
  - Existing methods for learning from demonstration
  - Focus on execution policies
- Adapt to collaborative setting
  - Move away from flat representations
  - Move away from divide-and-conquer planning mechanisms
  - Consider collaboration in a broad sense

Supportive Behaviors



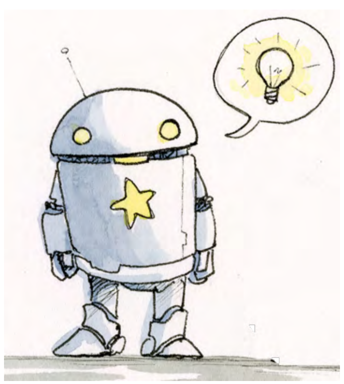
Can we do better than LfD-based Methods?



Demonstration-based Methods

Human figures out ***how*** and ***when*** the robot can be helpful

- \+ Quickly enables useful, helpful actions.
- − Does not scale with task count!
- − Requires human expert



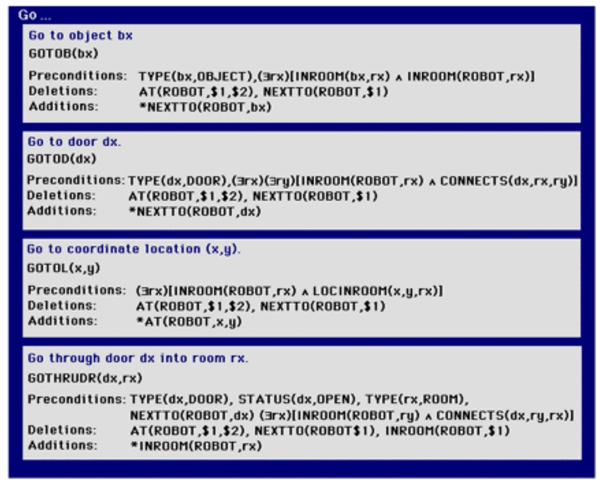
Planner-based Methods

Robot figures out ***how*** and ***when*** it can be helpful

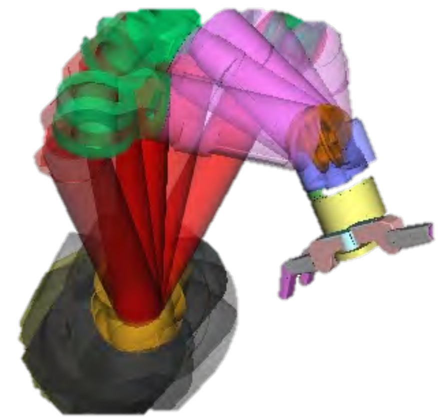
- Allows for novel behaviors to be discovered
- Enables deeper task comprehension and action understanding

Generating Supportive Behaviors

Perspective Taking



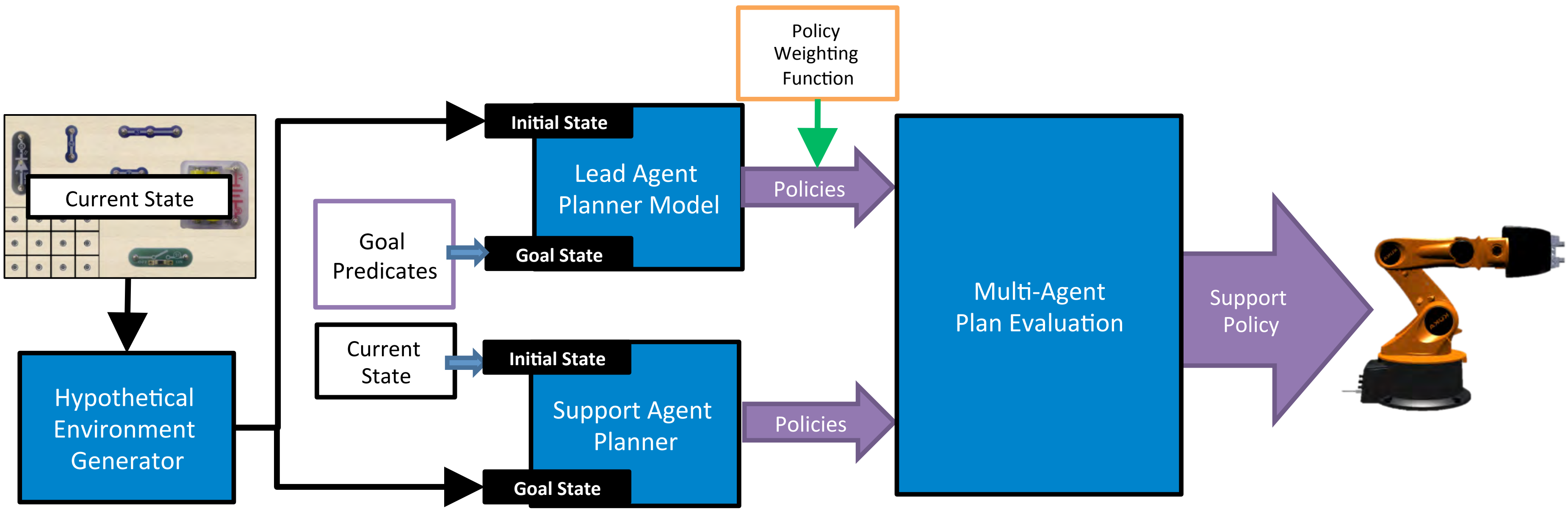
Symbolic planning



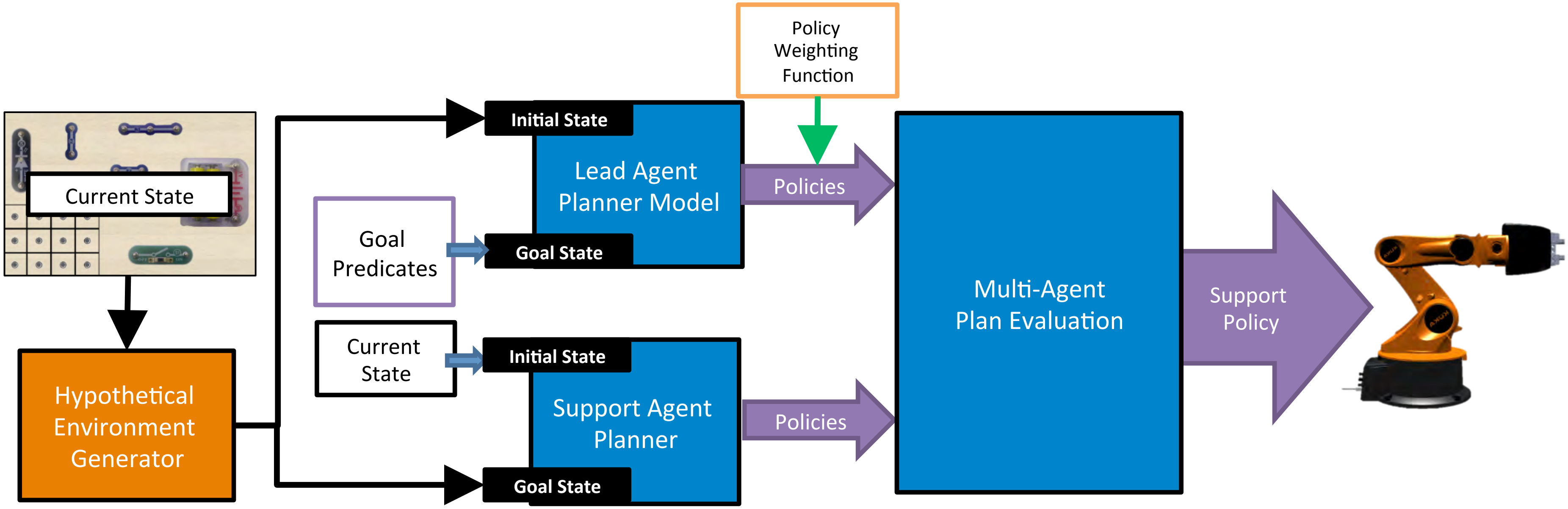
Motion planning

Autonomously Generated Supportive Behaviors

Supportive Behavior Planning

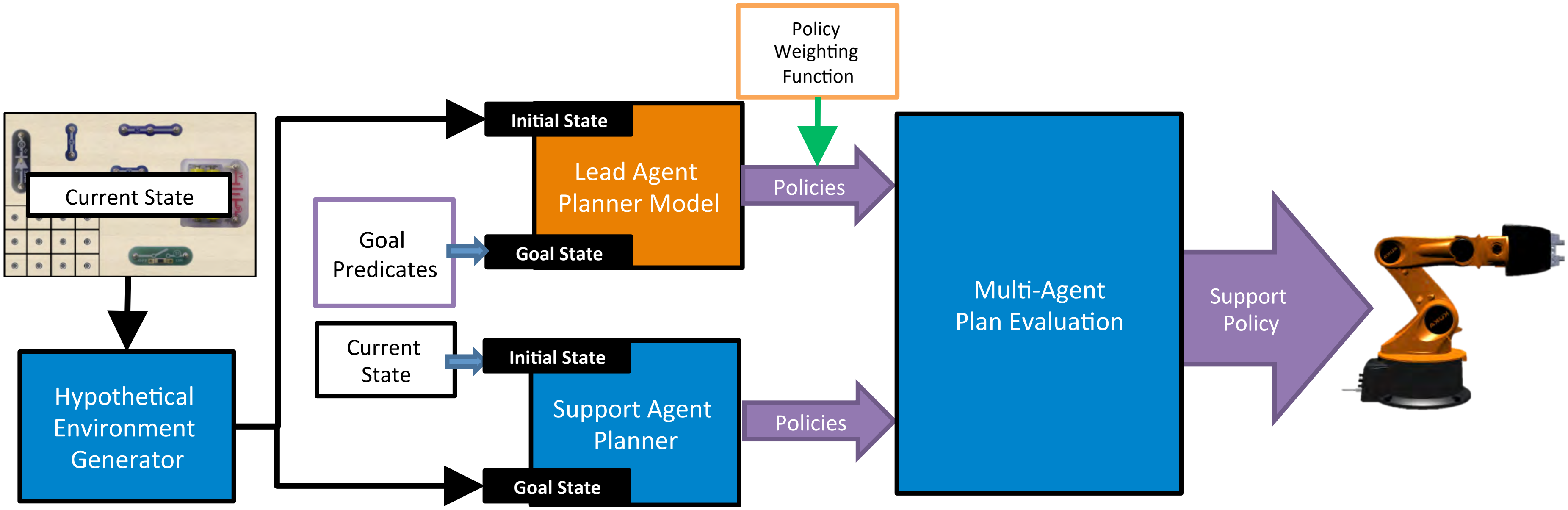


Supportive Behavior Planning



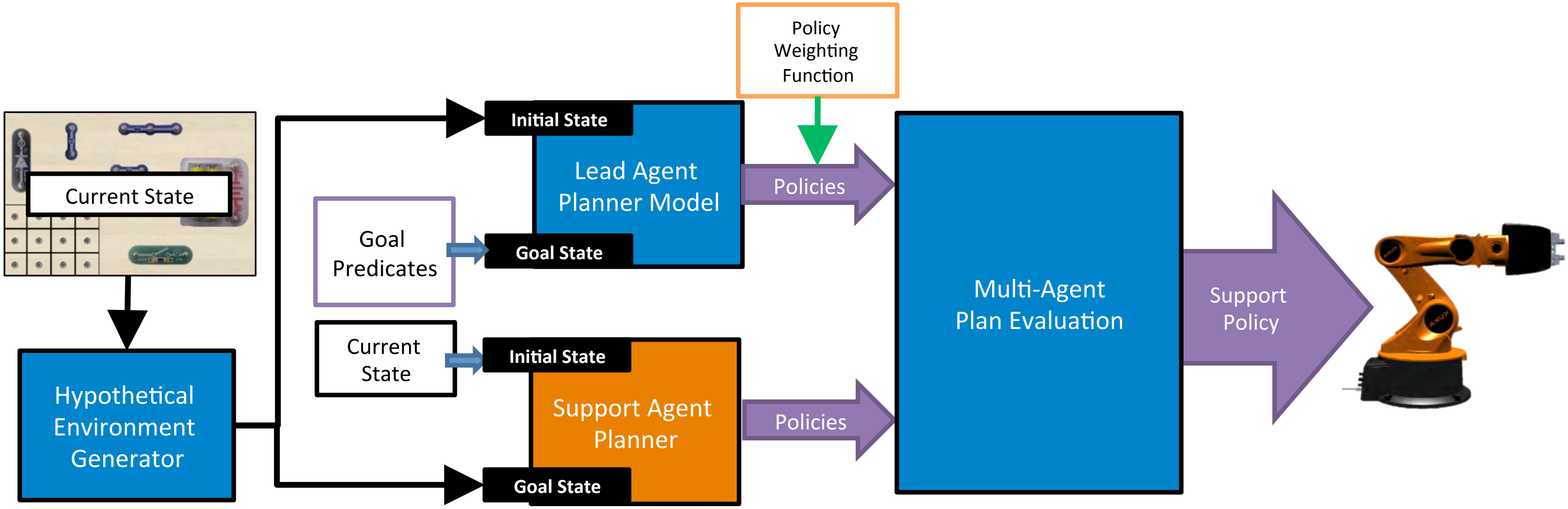
- Hypothesize future world states based on their plans

Supportive Behavior Planning



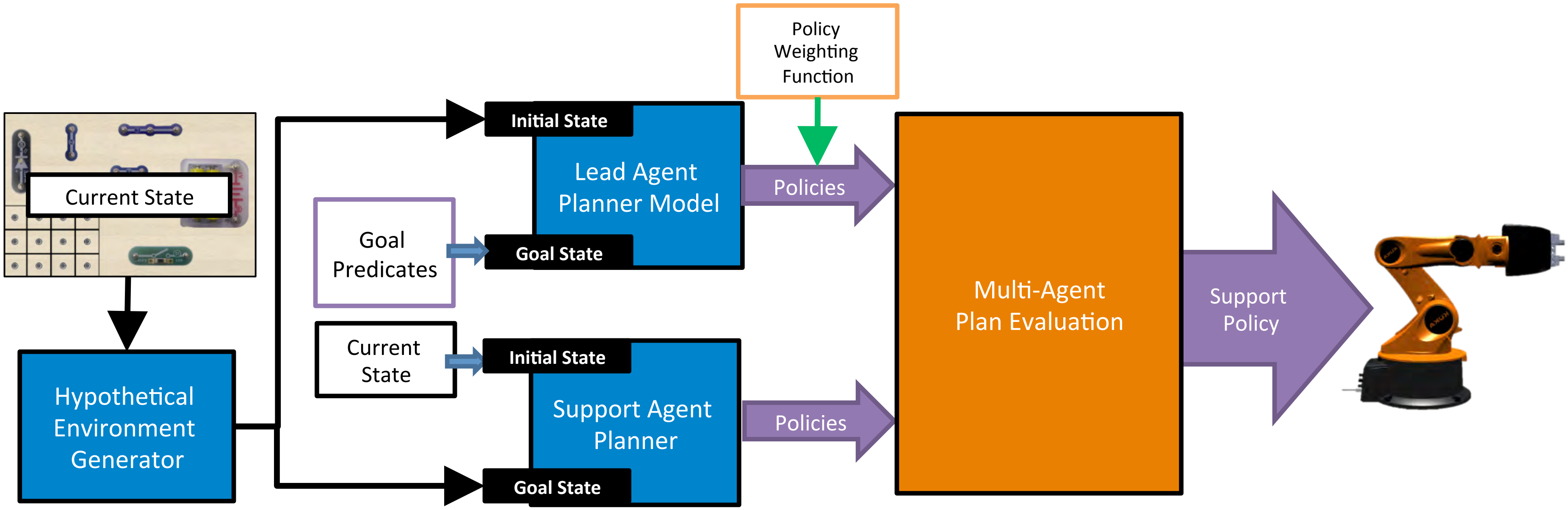
- Hypothesize future world states based on their plans
- Predict lead agent behavior using a user model

Supportive Behavior Planning



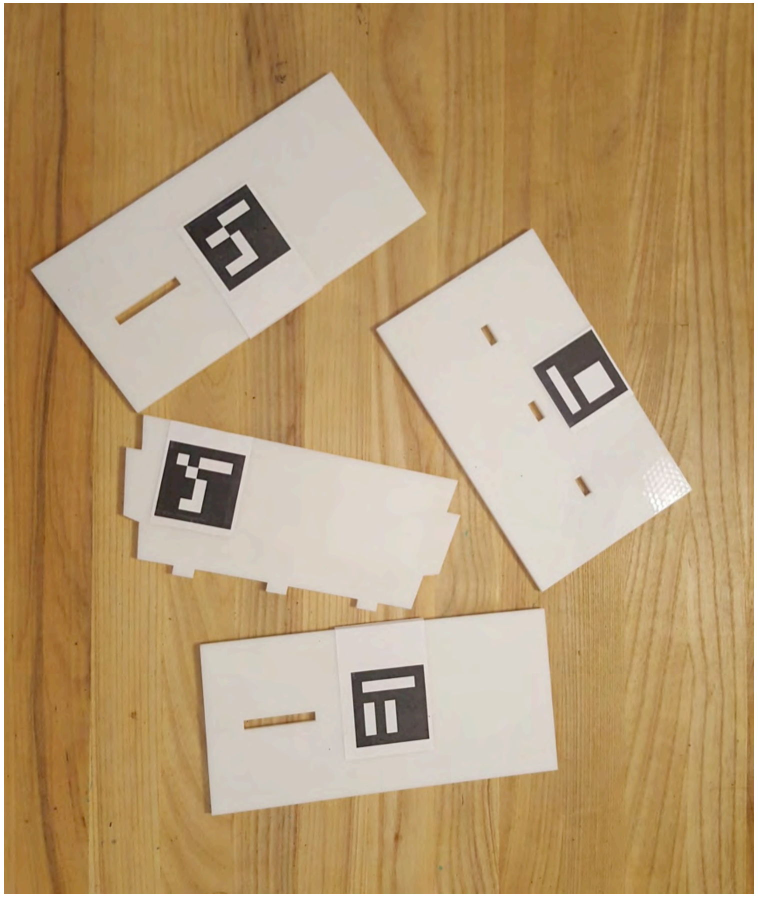
- Hypothesize future world states based on their plans
- Predict lead agent behavior using a user model
- Plan supportive actions that would simplify achieving this world state (or prevent sub-optimal plans)

Supportive Behavior Planning

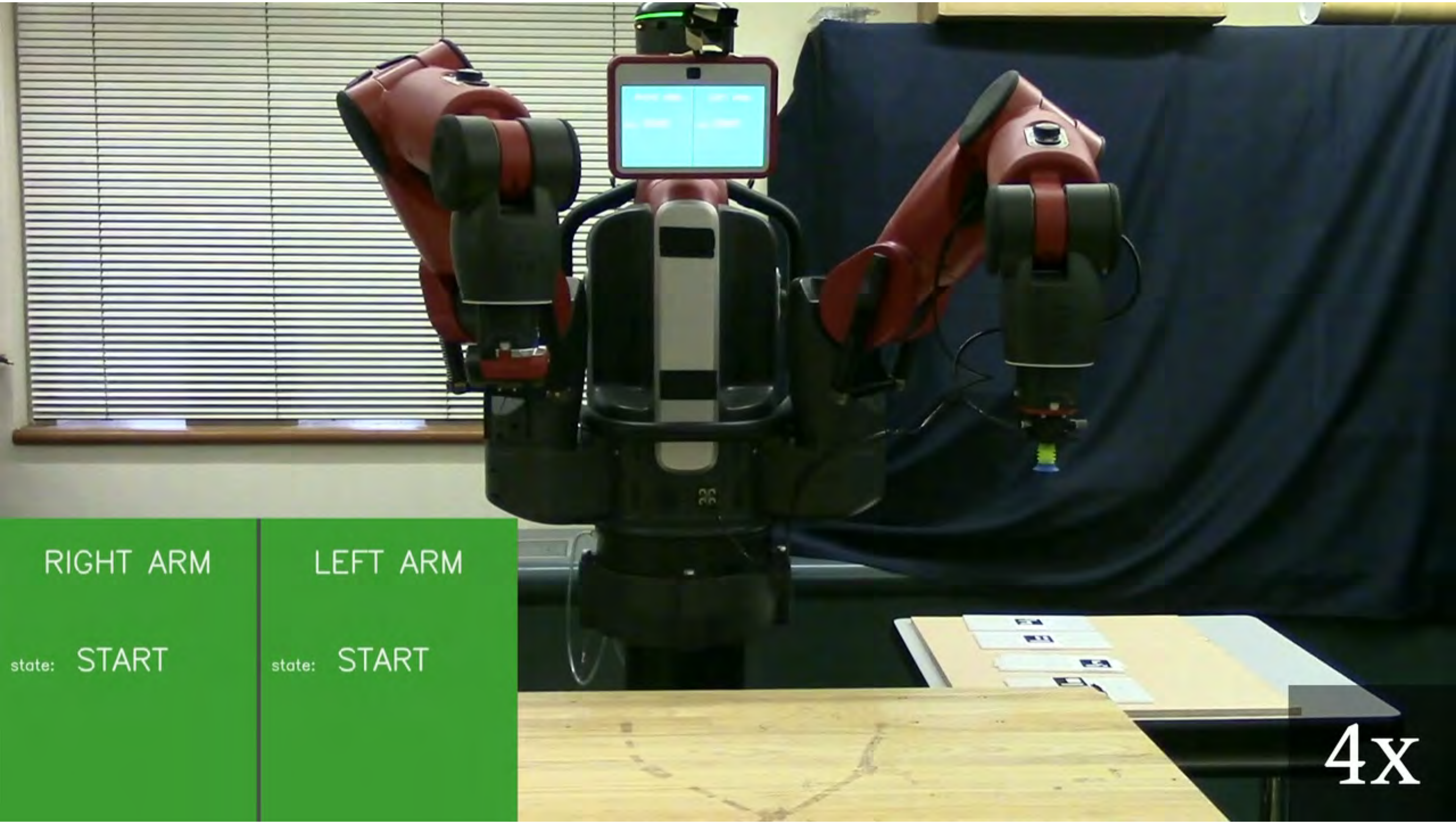


- Hypothesize future world states based on their plans
- Predict lead agent behavior using a user model
- Plan supportive actions that would simplify achieving this world state (or prevent sub-optimal plans)
- Evaluate multi-agent plan

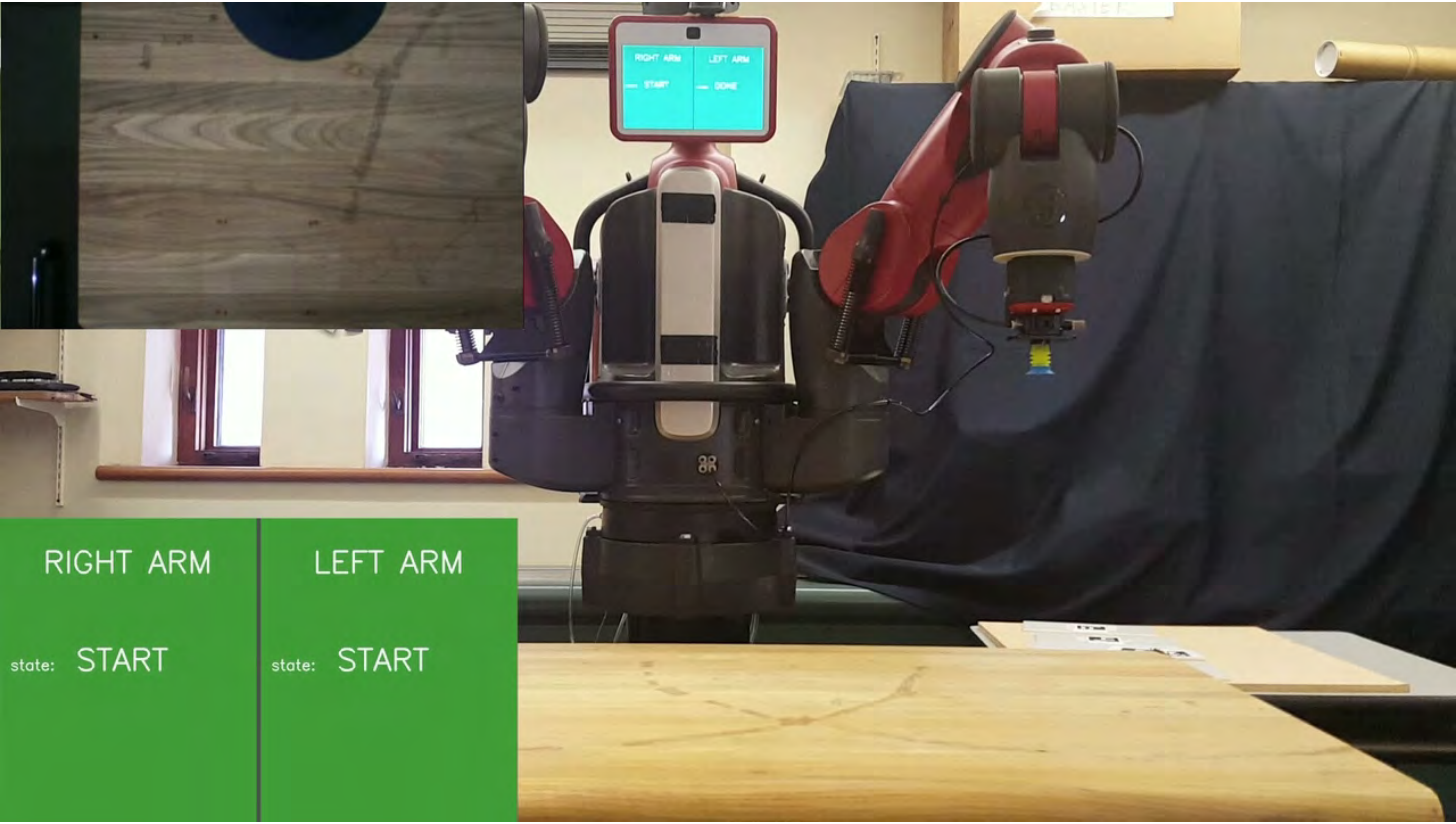
Simplified manipulation task



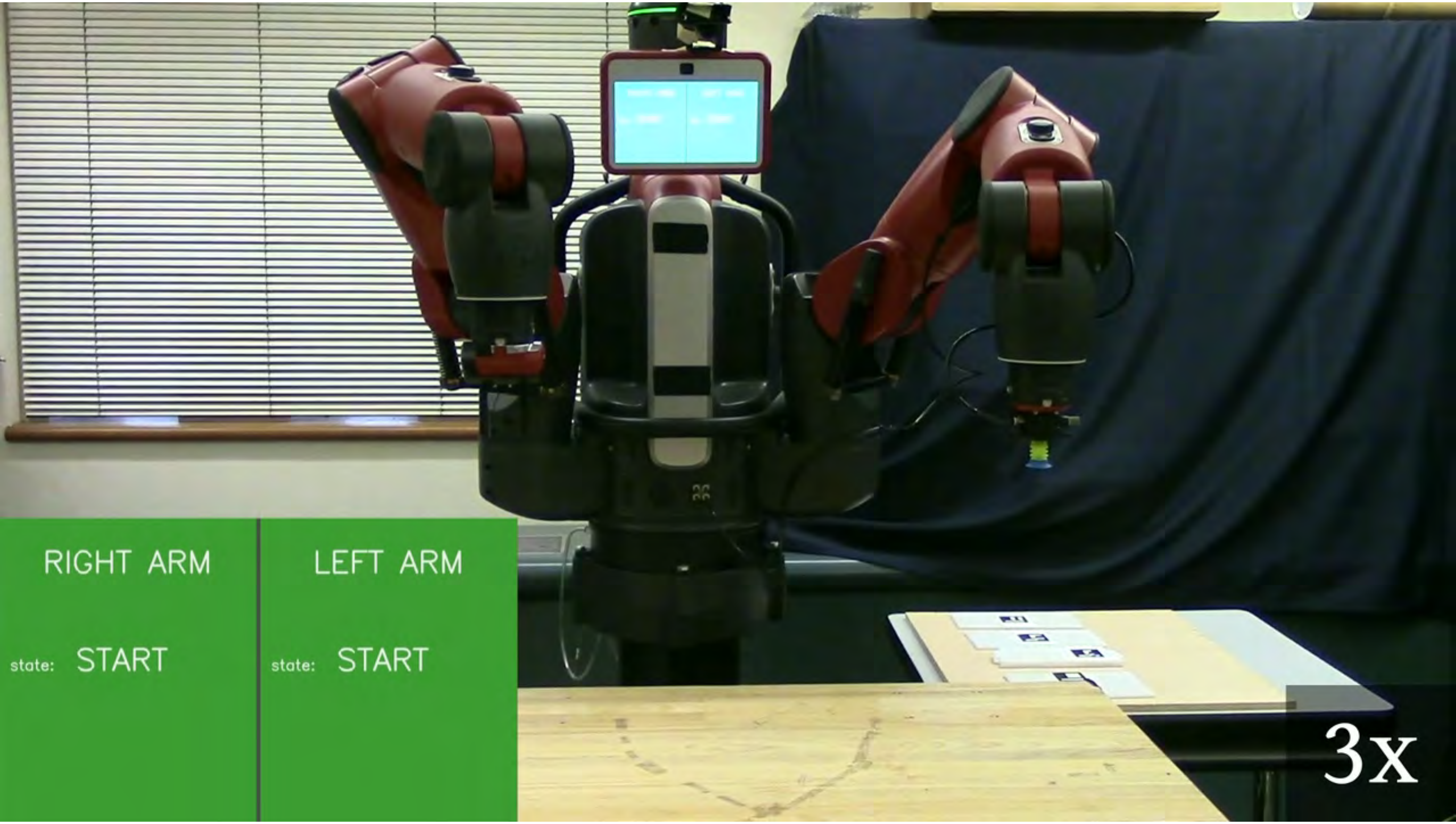
Supportive Action for Bench Assembly



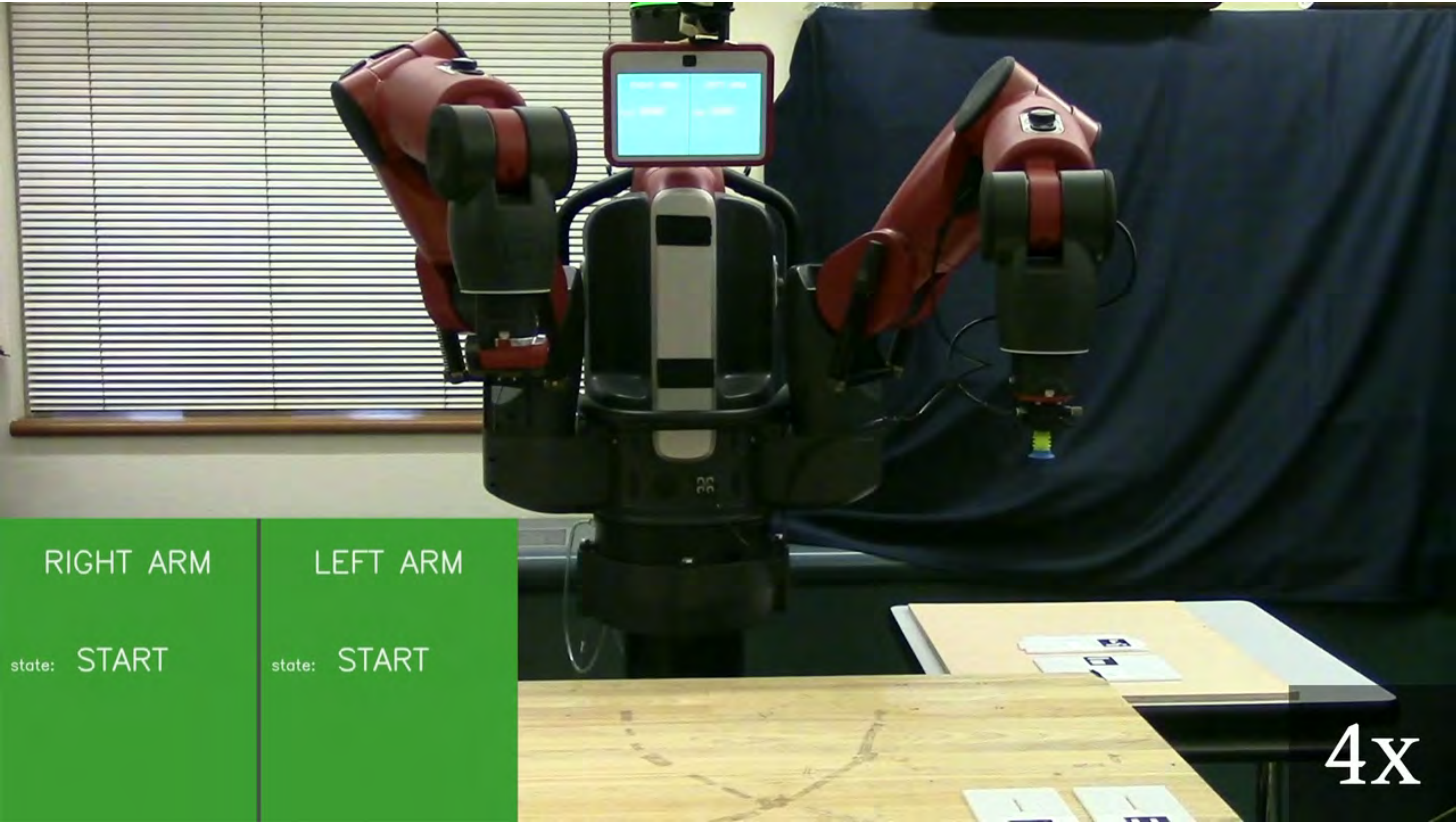
Simplified Vision, Control



Failure Recovery



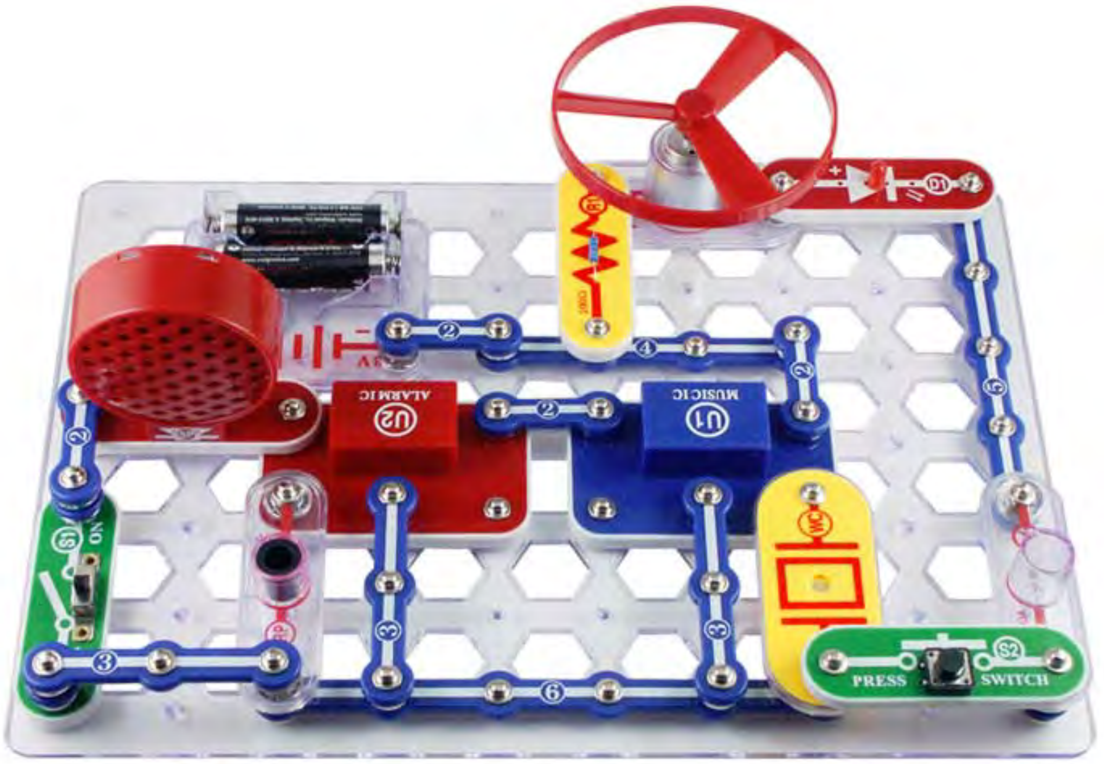
Preferences in Task Assignment



Changes as we consider Collaborative Manufacturing

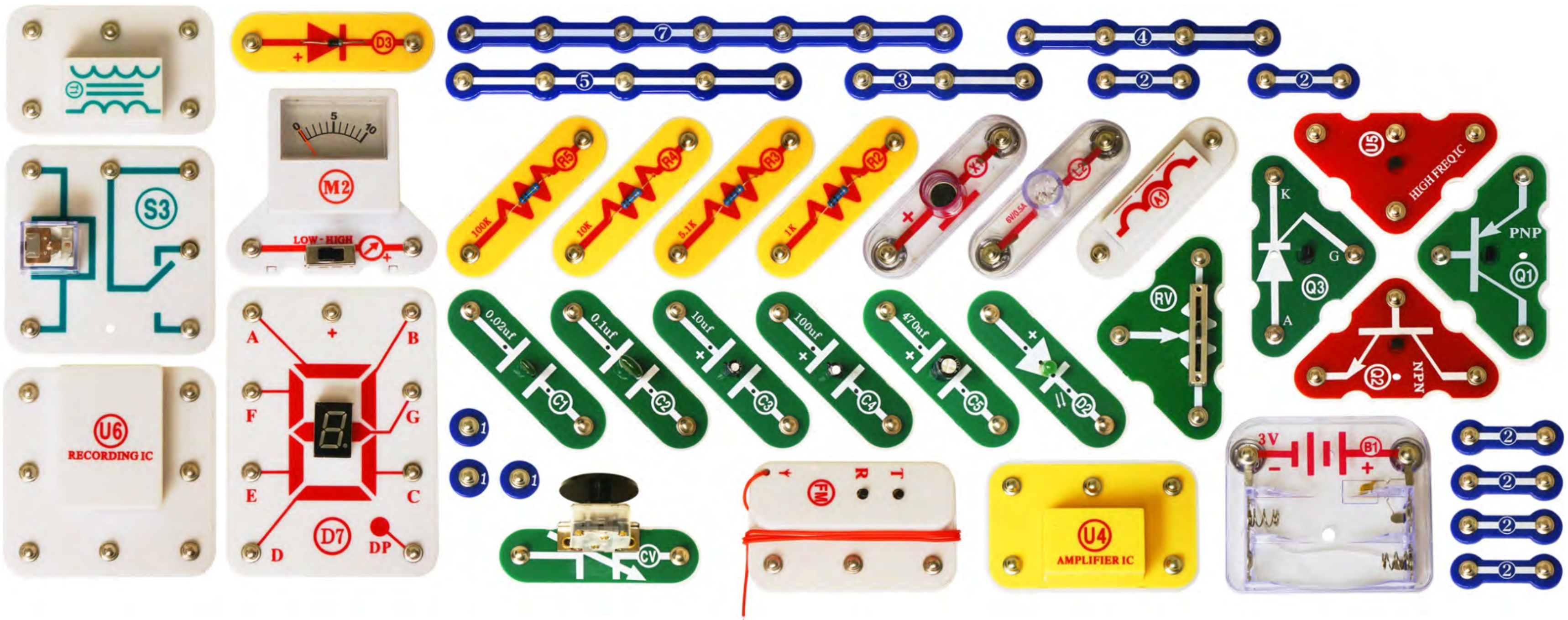
- Keep many of our pillars
  - Observations of sequential manipulation tasks
  - Existing methods for learning from demonstration
  - Focus on execution policies
- Adapt to collaborative setting
  - Move away from flat representations
  - Move away from divide-and-conquer planning mechanisms
  - Consider collaboration in a broad sense

Application Domain: SnapCircuits



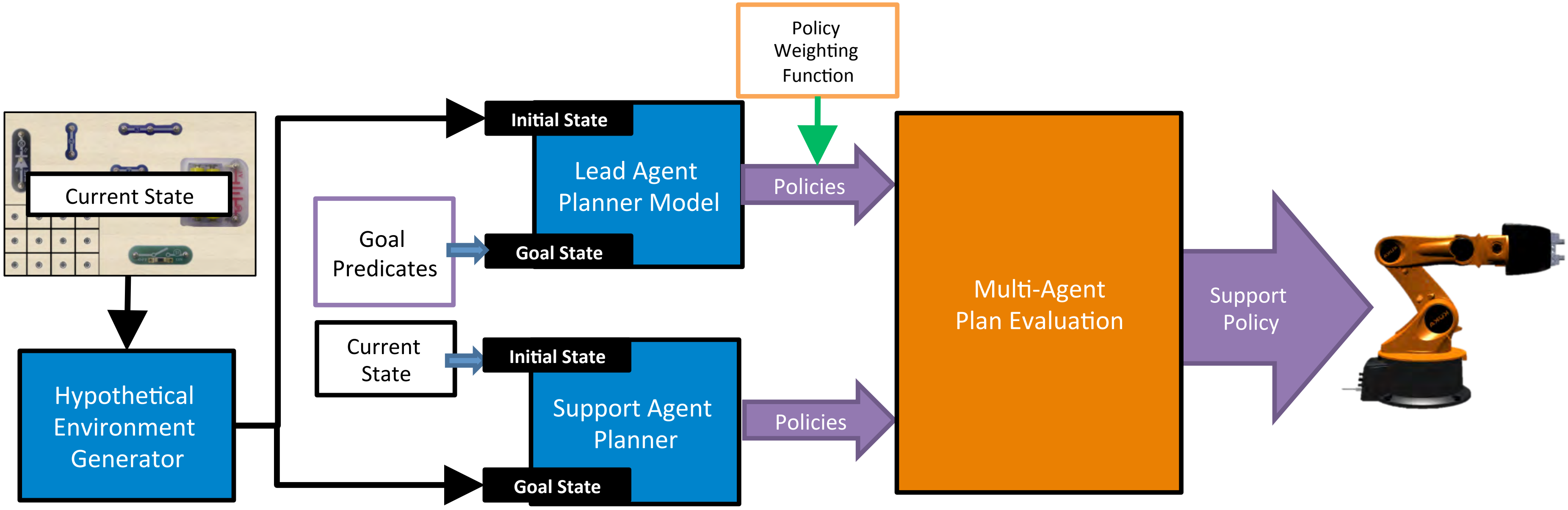
- “Construct a switched circuit with a power source and an LED”
- Many valid solutions
- Many suboptimal solutions exist
- Rigid, easily identified components

Application Domain: SnapCircuits



- Resource utilization
- Circuit board space utilization
- Role assignment
- Subtask parallelization

Supportive Behavior Planning



- Hypothesize future world states based on their plans
- Predict lead agent behavior using a user model
- Plan supportive actions that would simplify achieving this world state (or prevent sub-optimal plans)
- Evaluate multi-agent plan

Plan Evaluation

Choose the support policy (**ξ** ∈ **Ξ**) that minimizes the expected execution duration of the leader's policy (**π** ∈ **Π**) to solve the TAMP problem **T** from the current state ($\mathbf{s_c}$)

- Duration estimate must account for
  - Resource conflicts (shared utilization/demand)
  - Spatial constraints (support agent's avoidance of lead)

$$\min_{\xi \in \Xi} \sum_{\pi \in \Pi_T} w_\pi * \mathrm{duration}(T, \pi, \xi, s_c, \gamma)$$

Plan Evaluation

Choose the support policy (**$\xi \in \Xi$**) that minimizes the expected execution duration of the leader's policy (**$\pi \in \Pi$**) to solve the TAMP problem **T** from the current state (**s_c**)

- Duration esti
  - Resource … ation/demand)
  - Spatial co … nt's avoidance of lead)

Weighting function makes a big difference!

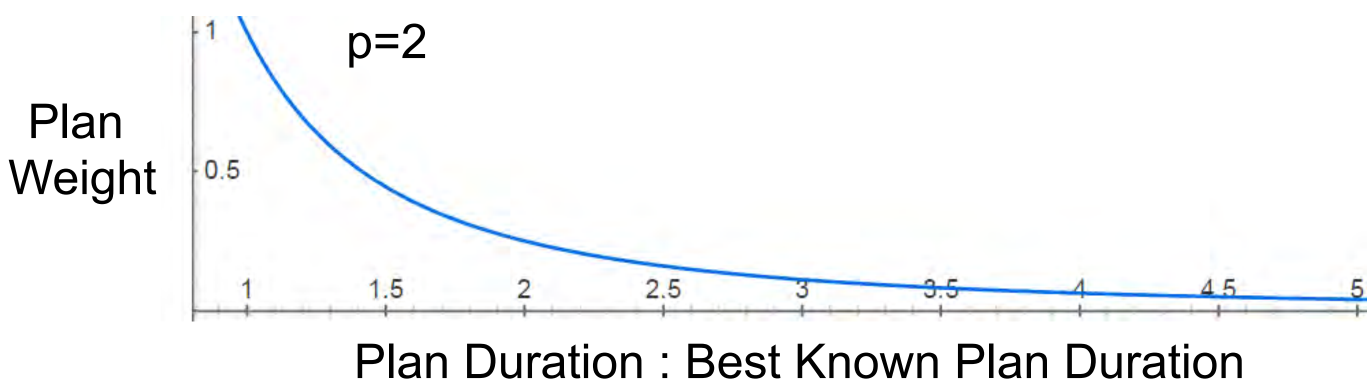
$$\min_{\xi \in \Xi} \sum_{\pi \in \Pi_T} w_\pi * \mathrm{duration}(T, \pi, \xi, s_c, \gamma)$$

Uniform Weighting Functions $w_\pi = 1$

Optimality-Proportional Weighting

$$w_\pi = \left(\frac{\min_{\pi \in \Pi_T} \text{duration}(T, \pi, \emptyset, s_0, f(x) = 1)}{\text{duration}(T, \pi, \emptyset, s_0, f(x) = 1)} \right)^{p}$$

Weight plans proportional to similarity vs. the best-known solution



Optimality-Proportional Weighting

Error Mitigation Weighting

$$w_\pi = \begin{cases} f(\pi) & ; \ \text{duration}(T, \pi, \emptyset, s_0, f(x) = 1) \leq \epsilon \\ -\alpha w_\pi & ; \ \text{otherwise} \end{cases}$$

Plans more optimal than some cutoff **ε** are treated normally, per *f*.

Suboptimal plans are negatively weighted, encouraging active mitigation behavior from the supportive robot.

$\alpha \leq \frac{1}{\max_\pi w_\pi}$ is a normalization term to avoid harm due to plan overlap

Error Mitigation Weighting

People who did all the work

Brad
Hayes

Alessandro
Roncone

Olivier
Mangin

Francesca
Stramandinoli

Thanks to...