Incorporating Planning and Reasoning into a Self-Motivated, Communicative Agent

Daphne Liu and Lenhart Schubert

Department of Computer Science
University of Rochester

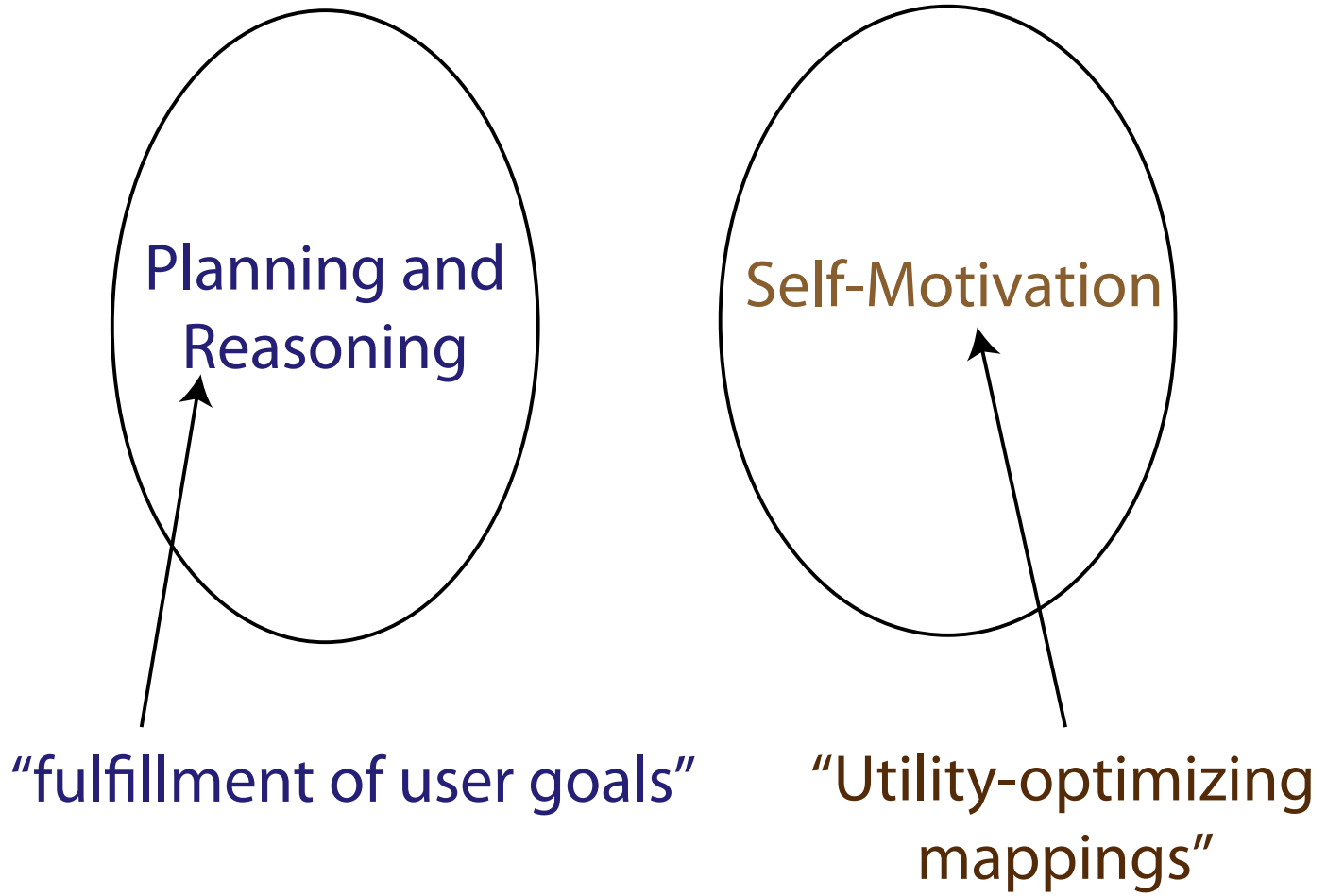
March 8, 2009

Outline

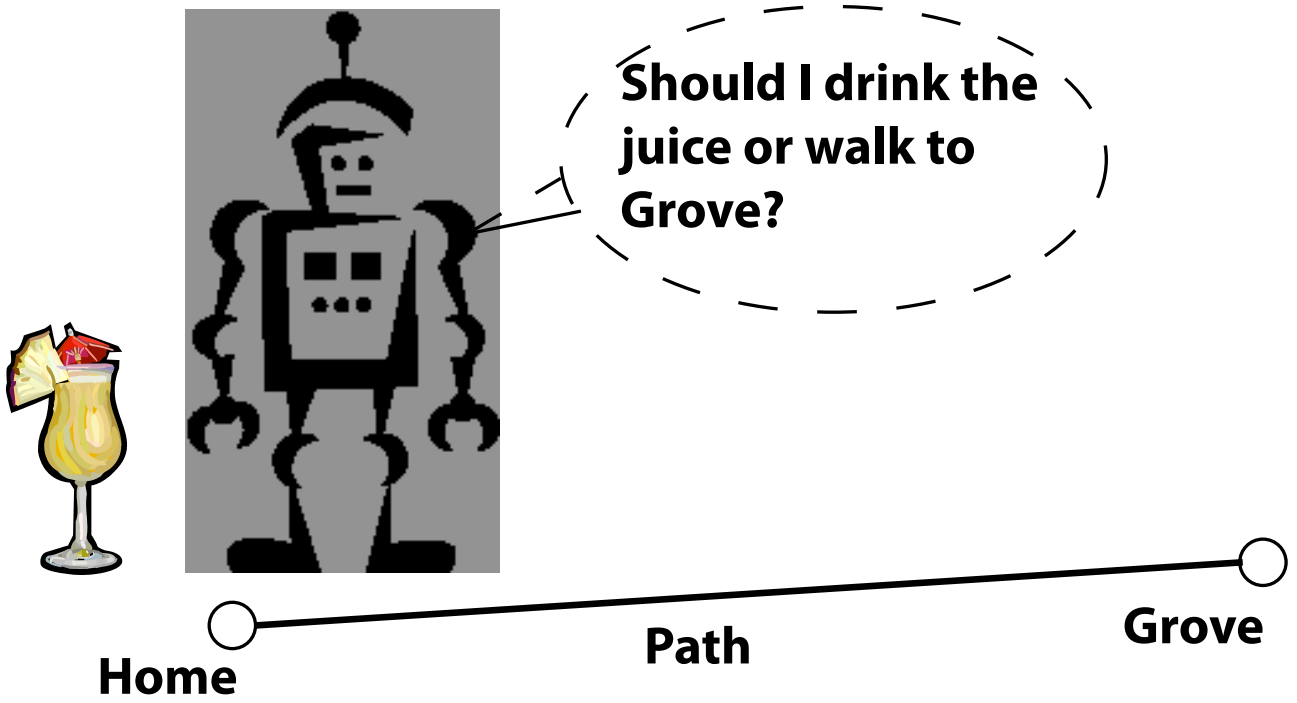
1. Motivation and Proposal
2. Architecture
3. Results
4. Conclusion

Planning, Reasoning, and Self-Motivation

How do we integrate them?

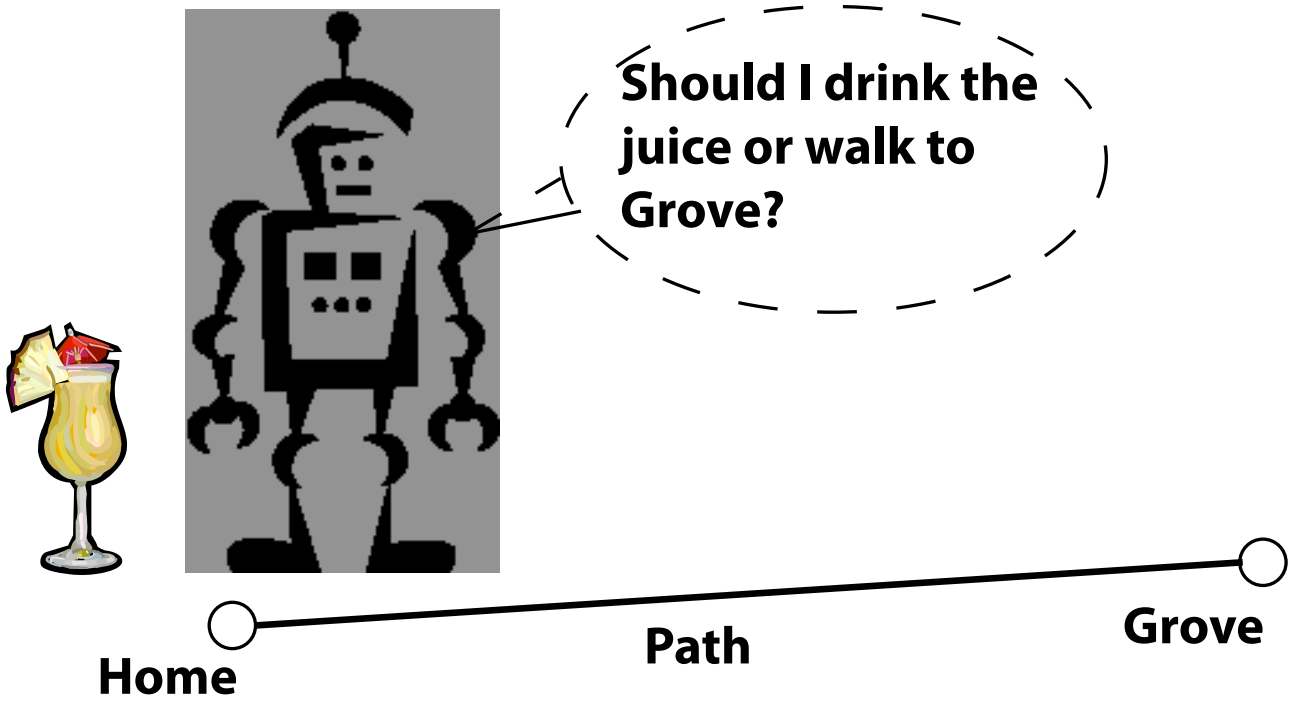


Our Proposal: Motivated Explorer (ME)



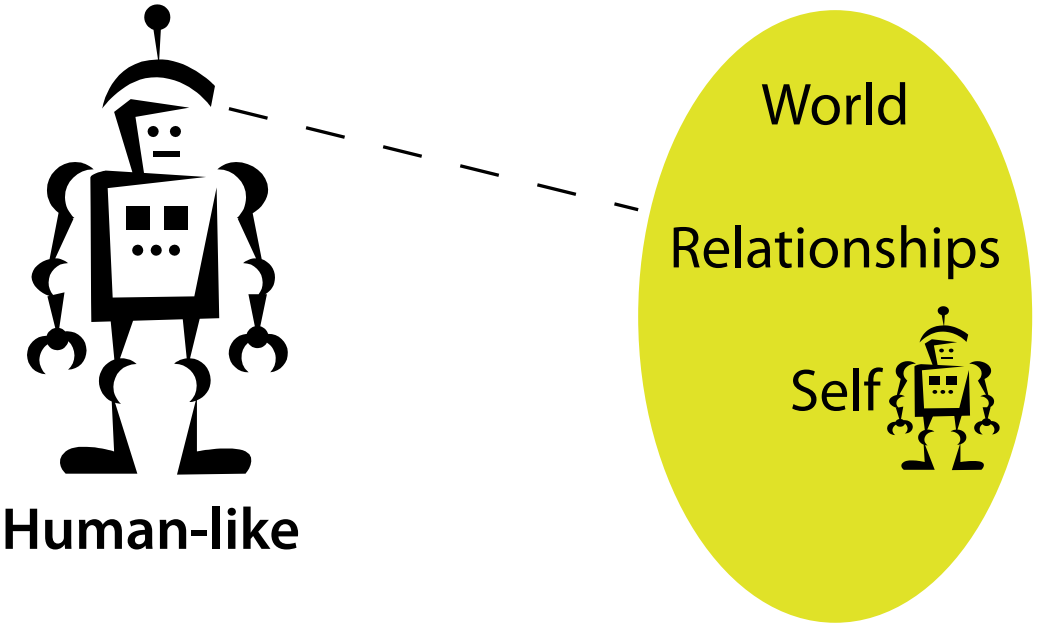
- Knowledge-based reasoning about actions and future states

Our Proposal: Motivated Explorer (ME)



- Knowledge-based reasoning about actions and future states
- Motivated by consideration of the long-range utility of choices

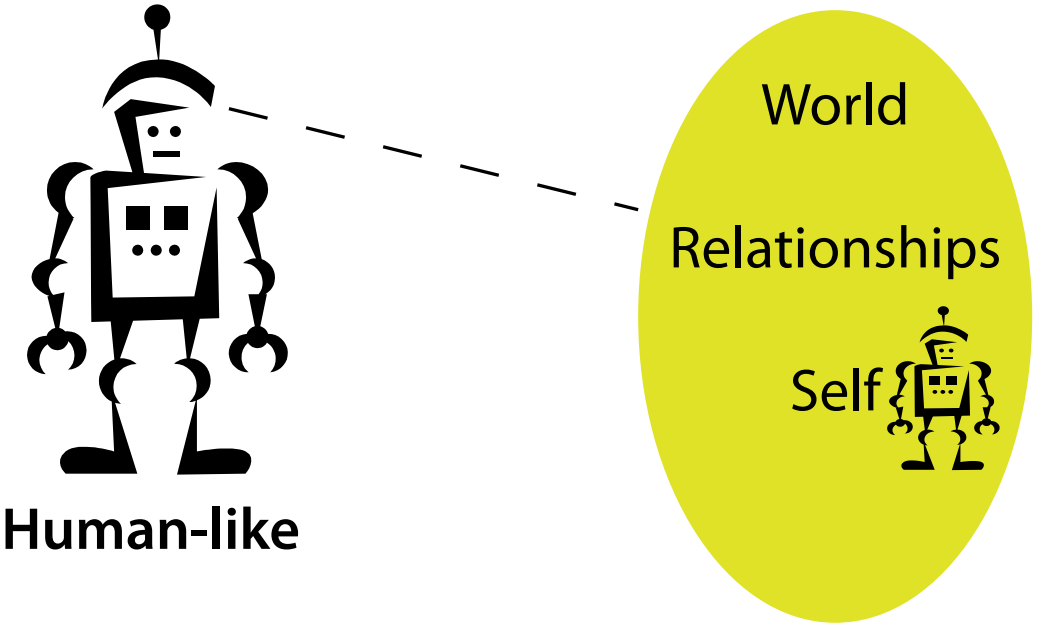
Explicit Self-Awareness



Explicit self-knowledge:

- amenable to self-observation and use

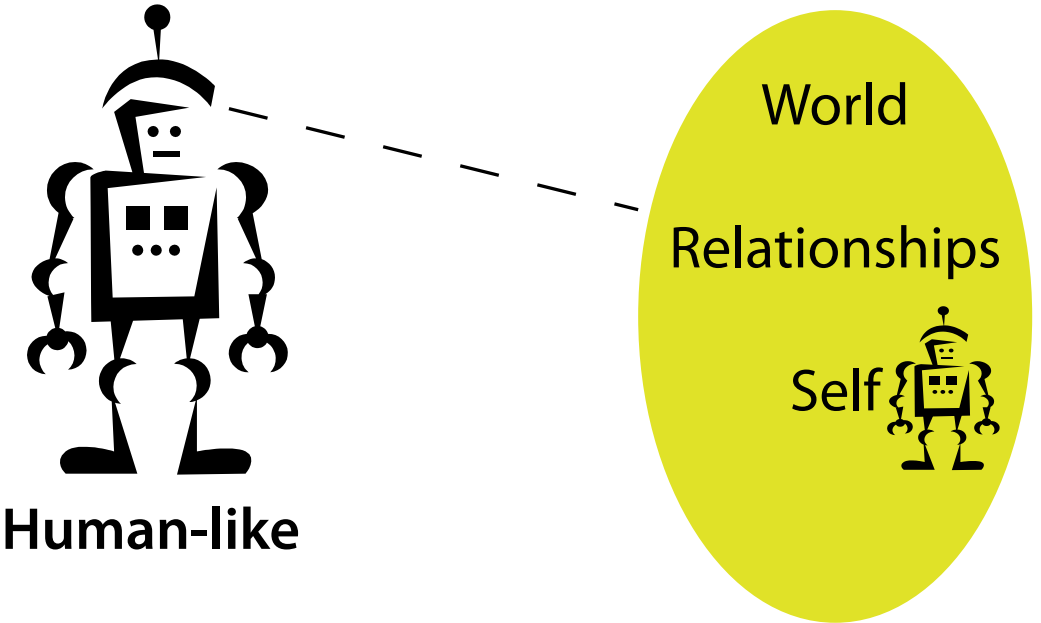
Explicit Self-Awareness



Explicit self-knowledge:

- amenable to self-observation and use
- conveyable by the agent

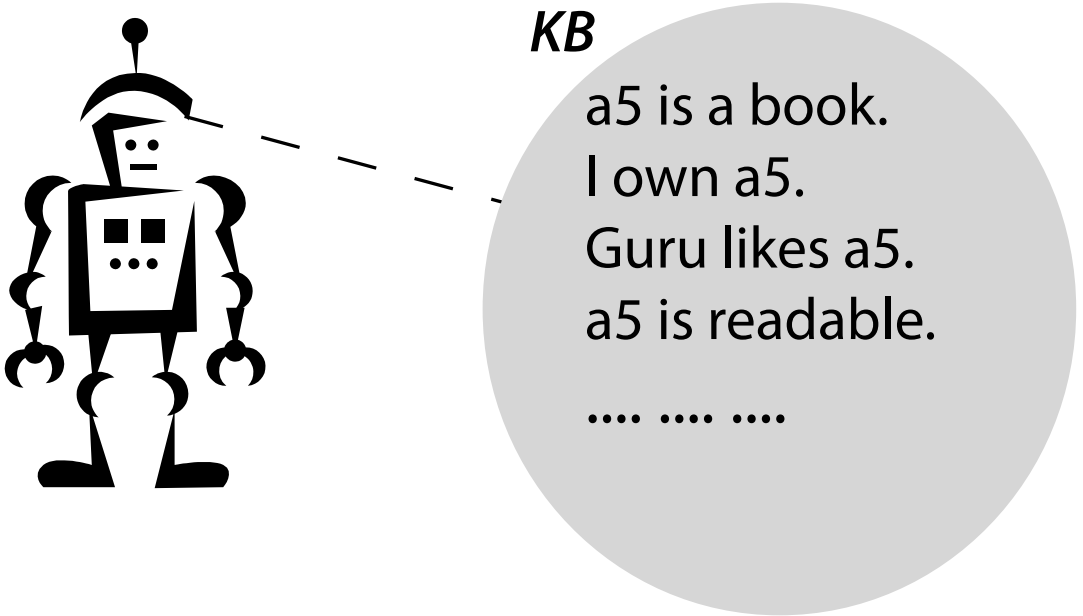
Explicit Self-Awareness



Explicit self-knowledge:

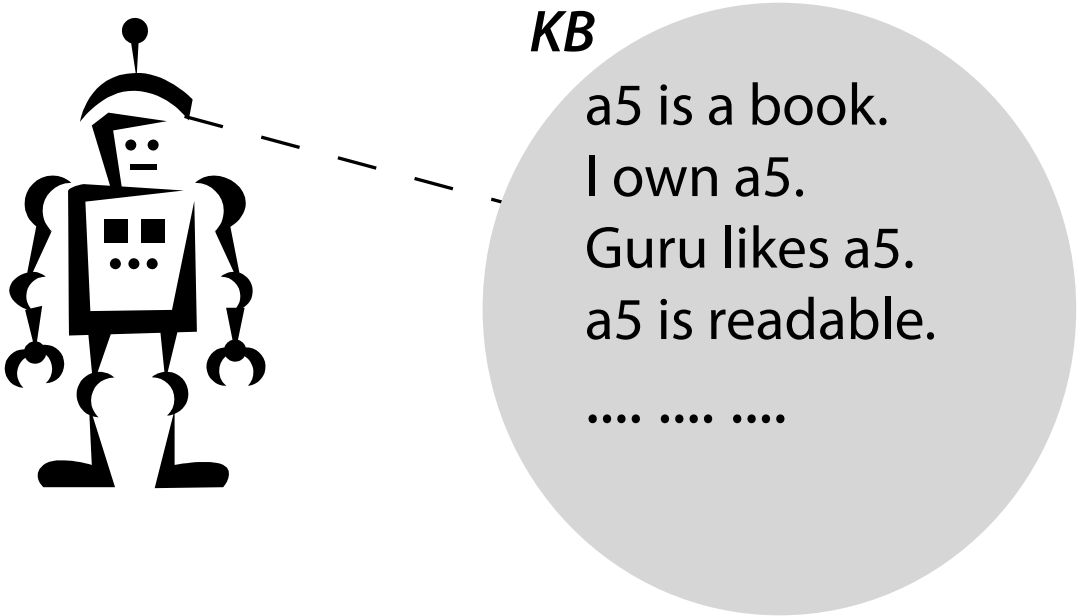
- amenable to self-observation and use
- conveyable by the agent
- open to inferences with world knowledge

ME's Explicit Self-Awareness



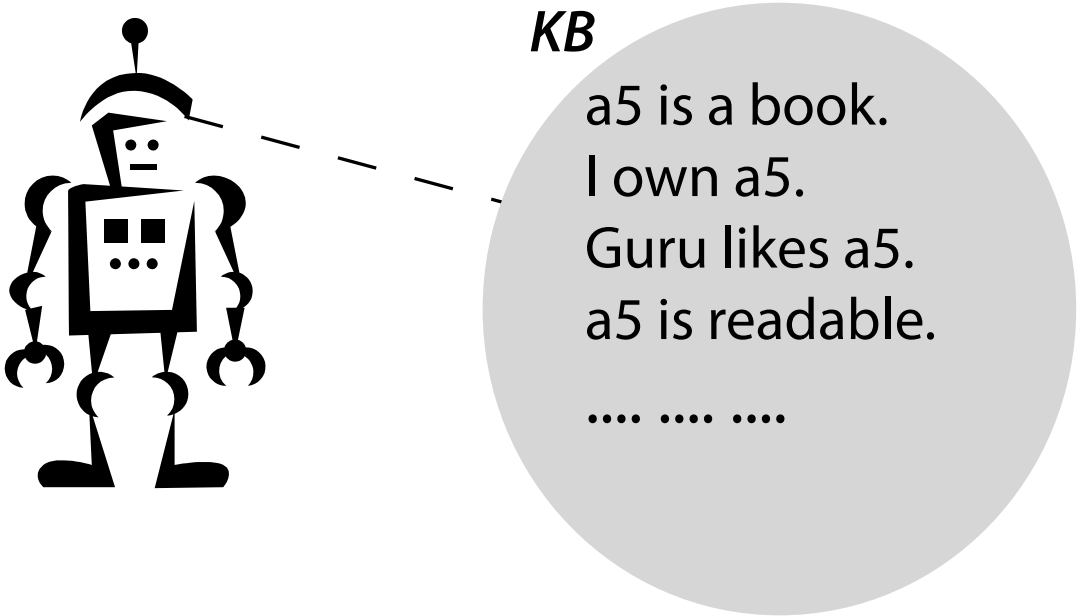
- Facts about itself, the current situation, and the world
- General knowledge in the form of Horn-like clauses
- Introspective:
  - Applicable operators and achievable goals

ME's Explicit Self-Awareness



- Facts about itself, the current situation, and the world
- General knowledge in the form of Horn-like clauses
- Introspective:
  - Applicable operators and achievable goals
  - Propositional attitudes

ME's Explicit Self-Awareness



- Facts about itself, the current situation, and the world
- General knowledge in the form of Horn-like clauses
- Introspective:
  - Applicable operators and achievable goals
  - Propositional attitudes
  - Actions and exogenous events so far

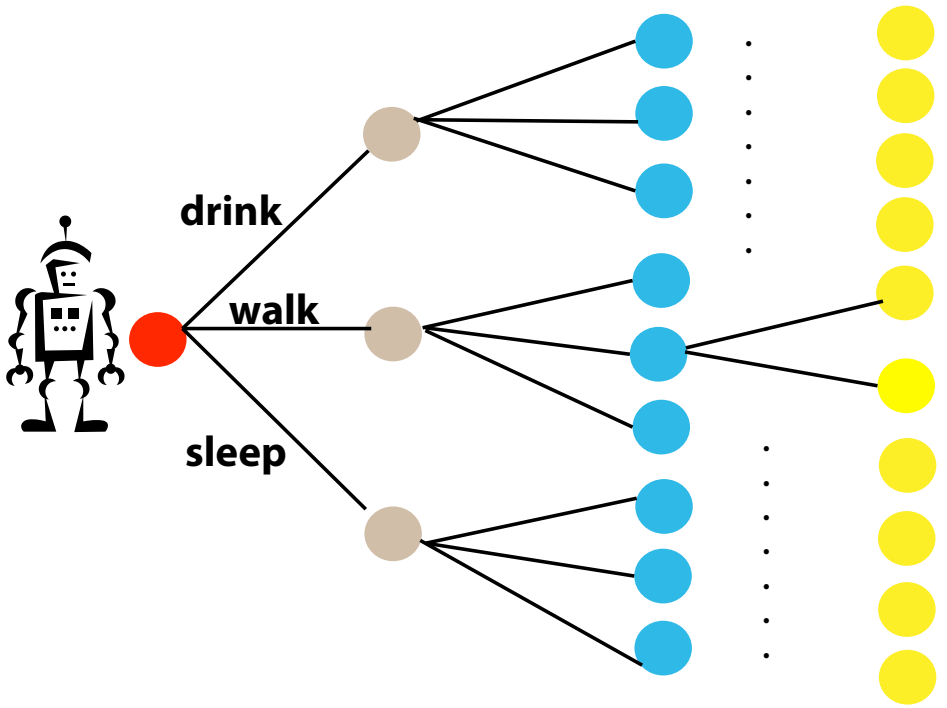
ME's Thoughtful Self-Motivation

- Driven by "desire" to maximize total utility, using own metrics of rewards and penalties

ME's Thoughtful Self-Motivation

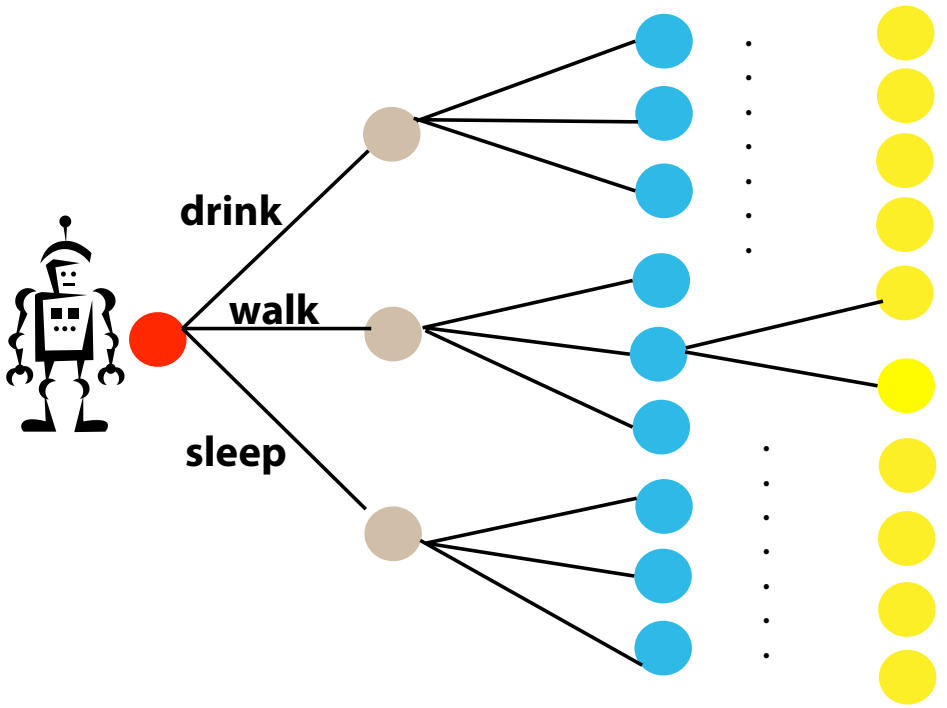
- Driven by "desire" to maximize total utility, using own metrics of rewards and penalties
- Grounded in reasoned lookahead and evaluation

Architecture: ME's Lookahead in Planning and Execution



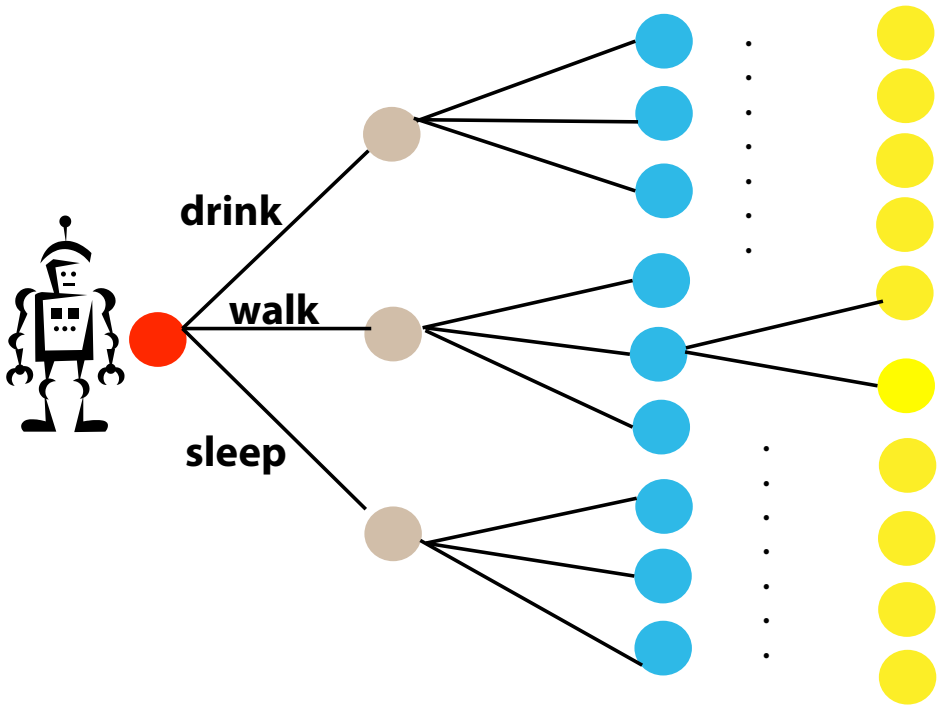
1. Search forward from a given state.

Architecture: ME's Lookahead in Planning and Execution



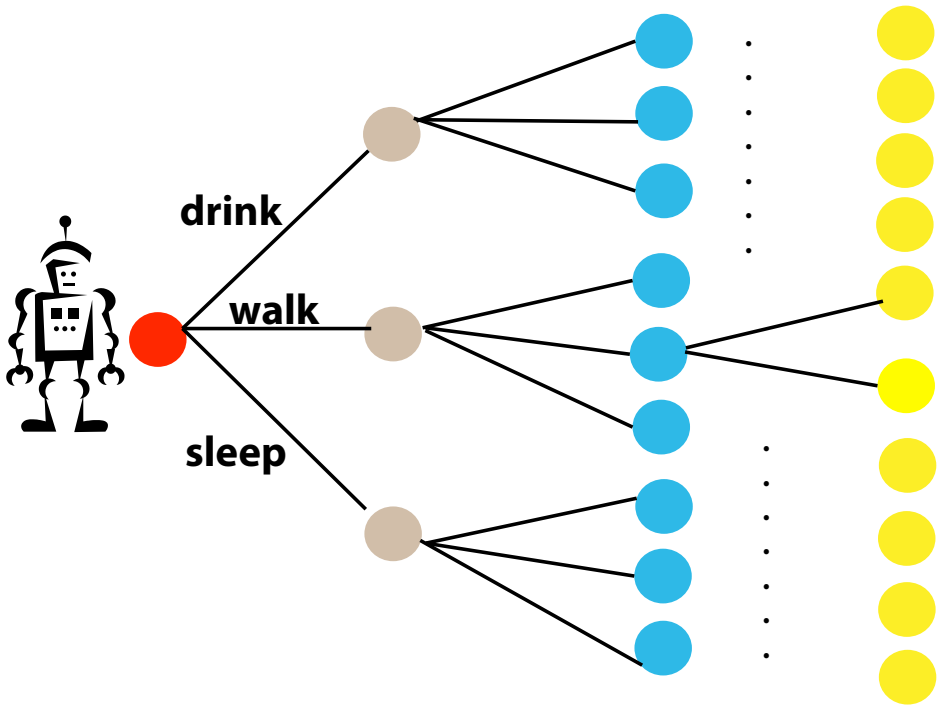
1. Search forward from a given state.
2. Propagate back expected rewards and costs of applicable actions and resulting states.

Architecture: ME's Lookahead in Planning and Execution



1. Search forward from a given state.
2. Propagate back expected rewards and costs of applicable actions and resulting states.
3. Execute the first action of the seemingly best plan.

Architecture: ME's Lookahead in Planning and Execution



1. Search forward from a given state.
2. Propagate back expected rewards and costs of applicable actions and resulting states.
3. Execute the first action of the seemingly best plan.
4. Update knowledge.

Architecture: Model vs. Actual Operators

- ME's incomplete knowledge of the world

Architecture: Model vs. Actual Operators

- ME's incomplete knowledge of the world
- Exogenous events (rain and fire)
  Example: A fire may start and disrupt ME's travel.

Architecture: Model vs. Actual Operators

- ME's incomplete knowledge of the world
- Exogenous events (rain and fire)
  Example: A fire may start and disrupt ME's travel.
- Multi-step actions

Architecture: Model vs. Actual Operators

- ME's incomplete knowledge of the world
- Exogenous events (rain and fire)
  Example: A fire may start and disrupt ME's travel.
- Multi-step actions
- The "actual" version of ME's chosen action is executed, updating ME's knowledge and the world.

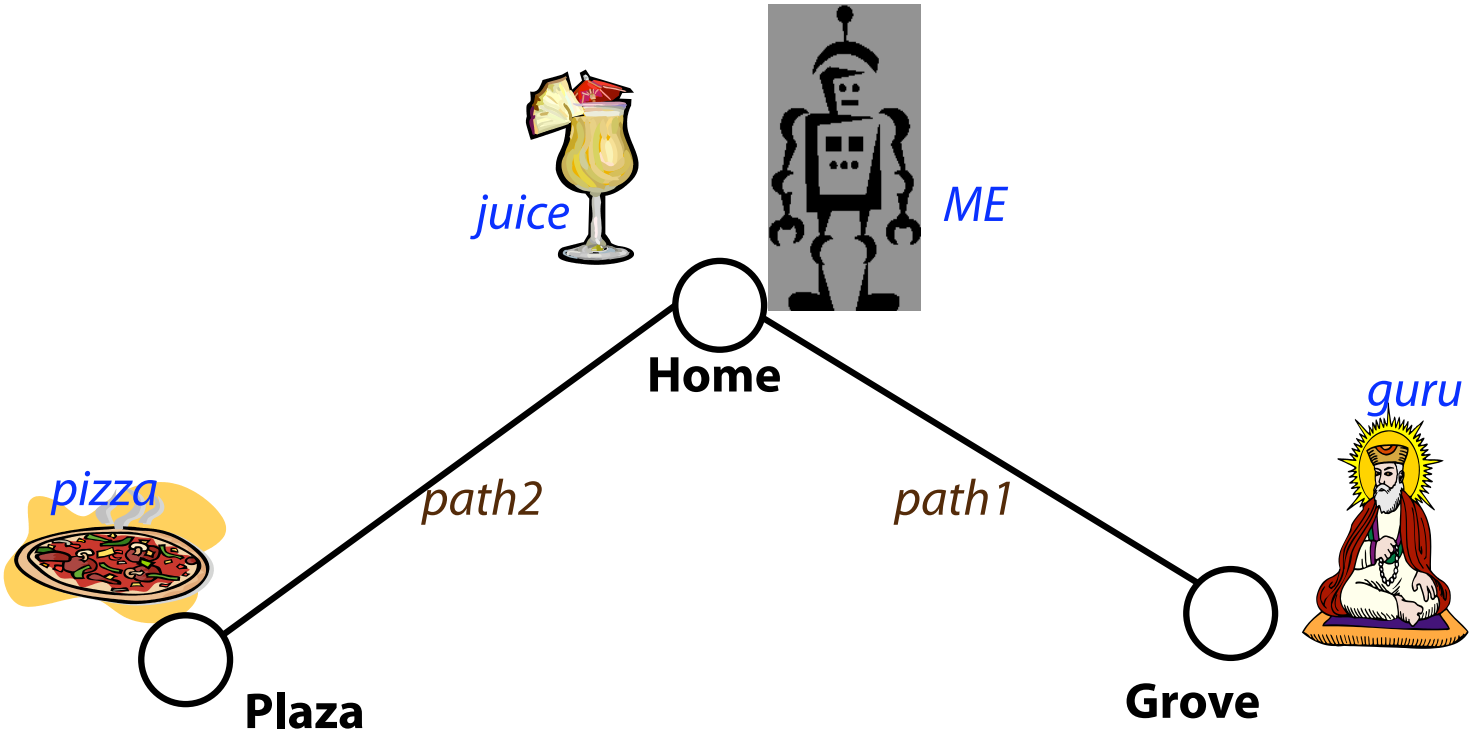
Example: Model Version of the Sleep Operator

```
(setq sleep (make-op
:name 'sleep
:pars '(?f ?h)
:preconds '((is_at ME home)  (is_tired_to_degree ME ?f)
                (>= ?f 0.5)  (> ?f ?h)  (not (there_is_a_fire))
                (is_hungry_to_degree ME ?h))
:effects '((is_tired_to_degree ME 0)
           (not (is_tired_to_degree ME ?f))
           (is_hungry_to_degree ME (+ ?h 2)))
:time-required '(* 4 ?f)
:value '(* 2 ?f)
))
```

Example: Actual Version of the Sleep Operator

```
(setq sleep (make-op
:name 'sleep.actual
:pars '(?f ?h)
:startconds '((is_at ME home) (is_tired_to_degree ME ?f)
                    (>= ?f 0.5) (> ?f ?h) (is_hungry_to_degree ME ?h))
:stopconds '((there_is_a_fire) (is_tired_to_degree ME 0))
:deletes '((is_tired_to_degree ME ?#1)
(is_hungry_to_degree ME ?#2))
:adds '((is_tired_to_degree ME (- ?f (* 0.5 (elapsed_time?))))
          (is_hungry_to_degree ME (+ ?h (* 0.5 elapsed_time?)))))
))
```

Simulated World



- *guru* knows whether *pizza* is edible.
- *ME* is thirsty and hungry, knows *juice* is potable and at home.
- Exogenous events: fire and rain
- Operators: walk, sleep, eat, drink, ask other agents whether something is true, answer the user's yes/no and wh- questions

Results of Goal-Directed Behavior

Ablation of Opportunistic Behavior

- *ME*'s sole goal: eating *pizza*
- Actions: asking *guru* to acquire food knowledge, traveling to reach *guru* ad *pizza*, and eating *pizza*
- Total utility of 66.5, after 18 steps

Output:

(RAIN 0), ((WALK HOME GROVE PATH1) 0),
((WALK HOME GROVE PATH1) 1), (FIRE 2),
((ASK+WHETHER GURU (EDIBLE PIZZA) GROVE)
3), (RAIN 5), ((WALK GROVE HOME PATH1) 5),
(FIRE 8), (RAIN 9), ((WALK GROVE HOME PATH1) 9),
((WALK HOME PLAZA PATH2) 12),
((WALK HOME PLAZA PATH2) 14), (FIRE 15),
((EAT PIZZA PLAZA) 17).

Results of Opportunistic Behavior

Opportunistic Behavior

- Total utility of 80.5, after 18 steps
- Direct result of seizing initial opportunity to drink *juice*

Output:

(RAIN 0), ((DRINK 4 JUICE HOME) 0), (FIRE 2), (RAIN 5), ((WALK HOME GROVE PATH1) 5), ((WALK HOME GROVE PATH1) 0) 6), (FIRE 7), ((ASK+WHETHER GURU (EDIBLE PIZZA) GROVE) 8), (RAIN 9), ((WALK GROVE HOME PATH1) 10), (RAIN 11), ((WALK GROVE HOME PATH1) 11), (RAIN 13), ((WALK HOME PLAZA PATH2) 13), ((WALK HOME PLAZA PATH2) 15), (RAIN 16), ((EAT PIZZA PLAZA) 17).

Question-Answering Example

>> (listen!)
You're welcome to ask ME a question.
((ask-yn user (can_talk guru)) (ask-wh user (is_animate ?y)))
==========

STEP TAKEN: (ANSWER_USER_YNQ (CAN_TALK GURU))

(GURU CAN TALK)
For question (CAN_TALK GURU), according to ME's currentknowledge base, ME offers the answer above.

STEP TAKEN: (ANSWER_USER_WHQ (IS_ANIMATE ?Y))

(ME IS ANIMATE)
(GURU IS ANIMATE)
For question (IS_ANIMATE ?Y), other than the above positive instance(s) that ME knows of, ME assumes nothing else as the answer.

Summary

- Explicitly self-aware and self-motivated *ME*

Summary

- Explicitly self-aware and self-motivated *ME*
- Deliberate self-motivation

Summary

- Explicitly self-aware and self-motivated *ME*
- Deliberate self-motivation
- Integration of behavioral and planning-based agents

Summary

- Explicitly self-aware and self-motivated *ME*
- Deliberate self-motivation
- Integration of behavioral and planning-based agents
- Towards a conversation agent with knowledge-and suggestion-driven dialogue behavior