

PhD Dissertation Defense
*Combining Self-Motivation with
Planning and Inference in a Self-Motivated
Cognitive Agent Framework*

Daphne Liu

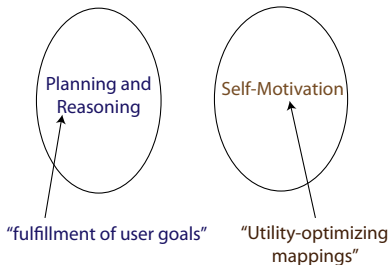
Dept. of Computer Science
University of Rochester

Dec. 12, 2012



HAJIM
SCHOOL OF ENGINEERING
& APPLIED SCIENCES
UNIVERSITY OF ROCHESTER

How do we integrate them?



Vision: Linguistically competent, intelligent, human-like agents

- 1 Bridge the planning & reasoning agent paradigm and the self-motivated agent paradigm.
- 2 Demonstrate the feasibility of combining planning, inference, and dialogue in a self-motivated cognitive agent.
- 3 Offer a versatile and easy-to-use self-motivated cognitive agent framework with competitive empirical results.

Self-Motivated Cognitive Agent Framework

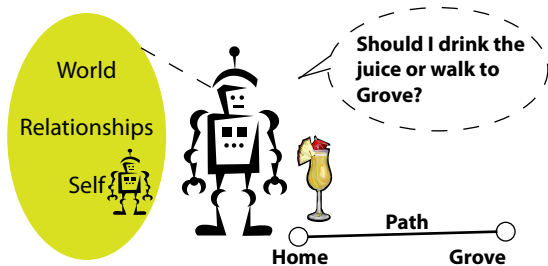
- Continual planning and self-aware reasoning aimed at optimizing long-term, cumulative rewards
- Planning treated as continually constructing, evaluating, and (partially) executing sequences of potential actions
- *Cognitive* system: ability to plan and reason with an expressively rich language

Design Open to User

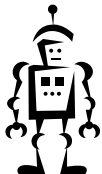
- User-designed actions and utility-measuring functions for actions and states
- User-specified “gridworld” roadmap placing entities at named locations with roads

High-Level Overview of Agent

Motivated Explorer (ME)



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



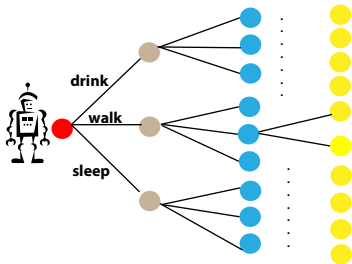
KB

a5 is a book.
I own a5.
Guru likes a5.
a5 is readable.
....

ME's Knowledge

- Facts about itself, the current situation, and the world
- General knowledge inference rules
- Capable of inferences and introspection

*Compared with the God's-eye view of the world,
ME's view may be incomplete, inaccurate or outdated.*



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.

Model vs. Actual Operators

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

How are the two versions used?

- ① *Model* version of ME's applicable actions contemplated in forward projection
- ② *Actual, stepwise* version of ME's chosen action 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?))))
))
```

Conveyance of Knowledge

>> (listen!)

You're welcome to ask ME a question.

```
((ask-yn user (guru can_talk))  
(ask-wh user (?y is_animate)))
```

=====

>> (go!)

STEP TAKEN: (ANSWER_USER_YNQ (CAN_TALK GURU))

GURU CAN TALK.

For question (CAN_TALK GURU), according to ME's current knowledge base, ME offers the answer above.

>> (go!)

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.

Use of (Restricted) Closed World Assumption

- Complete self-knowledge; *true* or *false*
- Relaxed CWA for a non-ME subject; *true*, *false*, or *unknown*

Restricted CWA

ME applies the CWA only for the two following cases:

- 1 literals about road connectivity and navigability; e.g., the absence of (*road path5*);
- 2 (a) when the subject is a local entity currently colocated with ME or one ME has visited, and (b) the predicate is non-occluded.

Types of Inference

- 1 Agent's knowledge in conjunction with general knowledge
- 2 Autoepistemic inferences
- 3 Epistemic inferences by simulative inference

Examples of General Inferences

Adding a rule to **general-knowledge**:

```
(push (list (list obj-type ?x) => (list property-i ?x)) *gen-knowledge*)
```

Definition of object types and respective properties:

```
(def-object 'expert '(is_animate can_talk))  
(def-object 'musical_instrument '(is_inanimate playable))
```

General inferences:

```
(all-inferences '[(expert guru), (musical_instrument piano)], *gen-knowledge*, *inf-limit*)  
=>  
(is_animate guru), (can_talk guru), (is_inanimate piano), (playable piano)
```

Simulative Inference Assumptions (only for animate entities)

- All AEs, like ME, have self-knowledge.
- All non-ME AEs are stationary.
- All AEs know of colocated objects, and all nonoccluded facts about such objects.

Examples of Autoepistemic and Simulative Inferences

Assumptions: *visited-objects* = {guru}, *occluded-preds* = {likes, knows}
//Autoepistemic Inferences

ACTION: (ANSWER_YNQ (NOT (IS_BORED ME)))

Answer: IT IS NOT THE CASE THAT ME IS BORED.

ACTION: (ANSWER_YNQ (CAN_FLY GURU))

Answer: IT IS NOT THE CASE THAT GURU CAN FLY.

ACTION: (ANSWER_YNQ (LIKES GURU PIZZA))

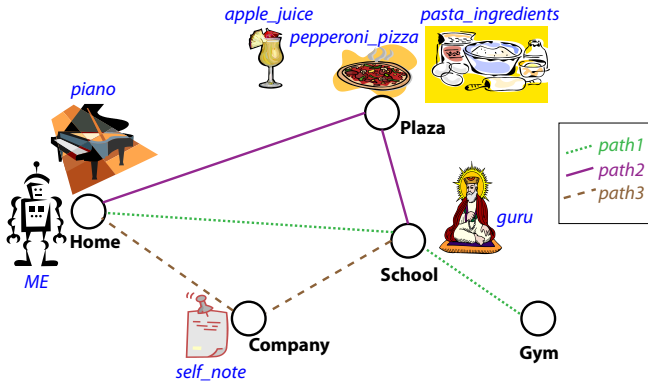
Answer: ME DOES NOT KNOW WHETHER GURU LIKES PIZZA.

//Simulative Inference

ACTION: (ANSWER_YNQ (KNOWS GURU
(WHETHER (LIKES GURU PIZZA))))

Answer: GURU KNOWS WHETHER GURU LIKES PIZZA.

Simulated World Example



- Exogenous fire and rain
- Operators: *walk*, *eat*, *drink*, *work_and_earn_money*, *buy*, *cook*, *swim*, *read*, *play*, *answer_user_yng*, *answer_user_whq*, *ask + whether*, *take_swimming_lesson*, *take_cooking_lesson*

Simulated World: A Goal-Directed Run

Sole Goal of Eating Self-Cooked Pasta

((WALK_HOME_SCHOOL_PATH1 0.0) 1 2 0)	((WALK_HOME_SCHOOL_PATH1 0.0) 2 2 1)
((TAKE_COOKING_LESSON 0.0 1.0) 1 4 3)	((TAKE_COOKING_LESSON 0.0 1.0) 4 4 6)
((TAKE_COOKING_LESSON 4.0 3.0) 1 4 8)	((TAKE_COOKING_LESSON 4.0 3.0) 4 4 11)
((TAKE_COOKING_LESSON 8.0 5.0) 1 4 13)	((TAKE_COOKING_LESSON 8.0 5.0) 4 4 16)
((TAKE_COOKING_LESSON 12.0 7.0) 1 4 18)	((TAKE_COOKING_LESSON 12.0 7.0) 4 4 21)
((TAKE_COOKING_LESSON 16.0 9.0) 1 4 23)	((TAKE_COOKING_LESSON 16.0 9.0) 4 4 26)
((WALK_SCHOOL_COMPANY_PATH3 11.0) 1 3 28)	((WALK_SCHOOL_COMPANY_PATH3 11.0) 3 3 30)
((WORK_AND_EARN_MONEY 4.0 0.0 12.5) 1 5 32)	((WORK_AND_EARN_MONEY 4.0 0.0 12.5) 5 5 36)
((WALK_COMPANY_SCHOOL_PATH3 17.5) 1 3 38)	((WALK_COMPANY_SCHOOL_PATH3 17.5) 3 3 40)
((WALK_SCHOOL_PLAZA_PATH2 19.0) 1 4 42)	((WALK_SCHOOL_PLAZA_PATH2 19.0) 4 4 45)
((BUY 15.0 PASTA_INGREDIENTS_PLAZA 2.0) 1 1 47)	
((WALK_PLAZA_HOME_PATH2 21.0) 1 2 49)	((WALK_PLAZA_HOME_PATH2 21.0) 2 2 50)
((COOK 6.5 20.0 22.0) 1 1 52)	
((EAT 6.5 PASTA) 1 1 54)	

Heuristic

- 1 Reward *eat*, *take_cooking_lesson*, *buy*, *cook*, and *work_and_earn_money*
- 2 Reward acquisition of cooking knowledge, money, pasta_ingredients, pasta; consumption of pasta or pasta_ingredients in states reached
- 3 Punish increase in hunger in states reached

Simulated World: An Opportunistic Run

```
((WALK HOME COMPANY PATH3 0.0) 1 2 0) ((WALK HOME COMPANY PATH3 0.0) 2 2 1)
((WORK_AND_EARN_MONEY 4.0 0.0 1.0) 1 5 3) ((WORK_AND_EARN_MONEY 4.0 0.0 1.0) 5 5 7)
((READ 6.5 SELF_NOTE COMPANY) 1 1 9)
((WALK COMPANY SCHOOL PATH3 6.0) 1 3 11) ((WALK COMPANY SCHOOL PATH3 6.0) 3 3 13)
((ASK+WHETHER GURU ((IS_POTABLE APPLE_JUICE) SCHOOL) 1 1 15)
((TAKE_COOKING_LESSON 0.0 7.5) 1 4 17) ((TAKE_COOKING_LESSON 0.0 7.5) 4 4 20)
((TAKE_COOKING_LESSON 4.0 9.5) 1 4 22) ((TAKE_COOKING_LESSON 4.0 9.5) 4 4 25)
((TAKE_COOKING_LESSON 8.0 11.5) 1 4 27) ((TAKE_COOKING_LESSON 8.0 11.5) 4 4 30)
((TAKE_COOKING_LESSON 12.0 13.5) 1 4 32) ((TAKE_COOKING_LESSON 12.0 13.5) 4 4 35)
((TAKE_COOKING_LESSON 16.0 15.5) 1 4 37) ((TAKE_COOKING_LESSON 16.0 15.5) 4 4 40)
((ASK+WHETHER GURU ((IS_EDIBLE PEPPERONI_PIZZA) SCHOOL) 1 1 42)
((WALK SCHOOL HOME PATH1 17.5) 2 2 45) ((WALK SCHOOL HOME PATH1 17.5) 1 2 44)
((PLAY 2.0 18.5 PIANO HOME) 1 1 47)
((SLEEP 19.0 7.0) 1 38.0 49) ((SLEEP 19.0 7.0) 38 38.0 86)
((WALK HOME SCHOOL PATH1 0.0) 1 2 88) ((WALK HOME SCHOOL PATH1 0.0) 2 2 89)
((WALK SCHOOL GYM PATH1 1.0) 1 1 91)
((TAKE_SWIMMING_LESSON 16.5 0.0 1.5 2.5) 1 3 93) ((TAKE_SWIMMING_LESSON 16.5 0.0 1.5 2.5) 3 3 95)
((TAKE_SWIMMING_LESSON 18.0 6.0 4.5 3.5) 1 3 97) ((TAKE_SWIMMING_LESSON 18.0 6.0 4.5 3.5) 3 3 99)
((TAKE_SWIMMING_LESSON 19.5 12.0 7.5 5.0) 1 3 101) ((TAKE_SWIMMING_LESSON 19.5 12.0 7.5 5.0) 3 3 103)
((WALK GYM SCHOOL PATH1 10.5) 1 1 105)
((WALK SCHOOL GYM PATH1 11.0) 1 1 107)
((WALK GYM SCHOOL PATH1 11.5) 1 1 109)
((WALK SCHOOL HOME PATH1 12.0) 1 2 111) ((WALK SCHOOL HOME PATH1 12.0) 2 2 112)
((WALK HOME PLAZA PATH2 13.0) 1 2 114) ((WALK HOME PLAZA PATH2 13.0) 2 2 115)
((BUY 15.0 PASTA_INGREDIENTS PLAZA 2.0) 1 1 117)
((WALK PLAZA HOME PATH2 14.0) 1 2 119) ((WALK PLAZA HOME PATH2 14.0) 2 2 120)
((COOK 21.0 20.0 15.0) 1 1 122)
((EAT 21.0 PASTA) 1 1 124)
((SLEEP 16.0 0.0) 1 32.0 126) ((SLEEP 16.0 0.0) 32 32.0 157)
((WALK HOME PLAZA PATH2 0.0) 1 2 159) ((WALK HOME PLAZA PATH2 0.0) 2 2 160)
((BUY 13.0 PASTA_INGREDIENTS PLAZA 2.0) 1 1 162)
((WALK PLAZA HOME PATH2 1.0) 1 2 164) ((WALK PLAZA HOME PATH2 1.0) 2 2 165)
((COOK 8.0 20.0 2.0) 1 1 167)
((EAT 8.0 PASTA) 1 1 169)
((WALK HOME PLAZA PATH2 3.0) 1 2 171) ((WALK HOME PLAZA PATH2 3.0) 2 2 172)
((BUY 11.0 APPLE_JUICE PLAZA 2.0) 1 1 174)
((DRINK 6.5 APPLE_JUICE) 1 1 176)
((WALK PLAZA HOME PATH2 4.0) 1 2 178) ((WALK PLAZA HOME PATH2 4.0) 2 2 179)
((WALK HOME COMPANY PATH3 5.0) 1 2 181) ((WALK HOME COMPANY PATH3 5.0) 2 2 182)
((WORK_AND_EARN_MONEY 0.0 9.0 6.0) 1 5 184) ((WORK_AND_EARN_MONEY 0.0 9.0 6.0) 5 5 188)
```

Additional opportunities seized: sleeping, playing piano, taking swimming lessons, gaining knowledge from reading and guru, eating & drinking foods other than pasta, working to earn more money

10 Runs of 40 Steps Each

- ① **Non-self-aware behavior:** average of -627.65
- ② **Goal-directed behavior (14 actions or 25 steps):**
average of 193.0
- ③ **Opportunistic behavior (3-step lookahead):**
average of 1260.85

Classical Planning: Towers of Hanoi

Challenges

- Effects not guaranteed to be persistent
- Rampant state duplication in forward search

Heuristic Function

- For placing *disk_j* on disk 3, reward = $j * (h - 1)$, where h = height of resulting “correct disk sequence”
- For removing *disk_j* from disk 3, symmetric penalty
- 0-reward *move* and 1-utility *do-nothing*

Results (averaged over 20 runs)

- 3-disk with 4-step horizon: optimal 7 steps taking 0.31s
- 4-disk with 8-step horizon: optimal 15 steps taking 55.35s

Classical Planning: Logistics

Domain

- 3 cities, each with an airport, a post office, and at least a truck
- 1 airplane

Heuristic Function

- Negative reward proportional to estimate of remaining distance to the goal state
- Negative reward for action failing to reduce estimated distance
- 0-utility for seemingly helpful actions, including *do-nothing*

Results

Solved problems requiring 3, 6, 9, 10, 13 steps in under 0.4s without missteps, with 2-step horizon

Continuous Planning: The Colorballs- $n-x$ Problem

Planning in Presence of Incomplete Info

Table: Working with or without Full Contingent Plans

	Contingent FF		Pond		CLG		SCAF	
	time	#acts	time	#acts	time	#acts	time	#acts
cb-4-1	0.27	277	0.98	102	0.35	295	6.31	22.18
cb-4-2	35.88	18739	40.92	1897	18.83	20050	8.70	36.14
cb-4-3	T		1063.11	28008	1537.99	1136920	11.72	45.14
cb-10-1	T		M		415.73	4445	313.89	246.94
cb-10-2	T		M		T		696.27	484.64

SCAF

- Actions: *walk* (ME's degree of happiness), *pick-up* (100), *put-down-color* (100), *announce-success* (100)
- No anticipated values for states
- 3-step horizon with branching factor 4, no heuristics

Continuous Planning: The Colorballs-9-*i* Problem

SCAF vs. Execution-Mode CLG

Problem	CLG in Execution Mode					
	Translation		Search		#acts	
	time	size (MB)	avg	max	avg	max
cb-9-1	20.9	16.5	1.21	7.80	33.7	197
cb-9-2	56.4	33.7	4.84	25.70	57.1	288
cb-9-3	113.7	51.4	46.26	122.19	76.3	367

Problem	SCAF			
	Run Time		#acts	
	avg	min / max	avg	min / max
cb-9-1	150.30	4.57 / 516.61	168.5	5 / 543
cb-9-2	281.38	16.37 / 642.90	239.12	11 / 552
cb-9-3	345.33	62.60 / 799.17	333.58	51 / 694

SCAF

- No translation needed and file size under 15 KB;
As *i* increases, an additional (*place-object...*) suffices
- Meandering actions and repeatedly visiting same states

Background (Brenner & Nebel, 2009)

- Each agent as an independent MAPSIM process
- No inter-agent communication, coordination or collaboration

Multiagent SCAF

- Coexisting agents, each with its own kb, etc. but sharing the world
- Actions: *walk* (10 if goal location, 0 otherwise), *stay-put* (10 if goal location, -1 otherwise)
- No anticipated values for states
- 4-step horizon; no heuristics

Continuous Multiagent Planning: Multiagent- $n-x-b$

Problem	SCAF			
	Run Time		#acts	
	avg	min / max	avg	min / max
ma-6-4-10	15.63	2.19 / 63.10	70.86	10 / 288
ma-10-1-15	64.61	0.93 / 397.89	80.28	1 / 519
ma-10-2-15	112.56	3.22 / 705.74	143.62	4 / 945
ma-10-3-15	160.32	6.27 / 772.98	202.26	7 / 907
ma-10-4-15	239.05	13.37 / 628.09	280.18	15 / 773
ma-10-5-15	282.47	16.5 / 878.35	358.92	20 / 1162
ma-10-6-15	351.49	37.79 / 1021.05	366.84	39 / 1038
ma-10-7-15	491.94	82.75 / 1531.86	498.02	88 / 1658

Brenner & Nebel's Results

- Successful run iff all agents reached their goals within 10 minutes
- No absolute rates, but normalized *relative* to the full visibility case
- Agents seeing only immediately adjacent locations with relative rate 37% - 62%
=> many runs failed

SCAF Discussion

- Average run times well under 10 minutes
- Impressive SCAF results, considering B&N's HTN-like technique

- Vere & Bickmore's Homer
- Winograd's SHRDLU
- Shapiro's GLAIR/Cassie
- TRIPS by Allen, Ferguson et. al

Summary of Contributions

- *Integration of self-motivation with planning & reasoning:*
epistemic inference, incomplete knowledge, continuous planning, question-answering, and cumulative utility optimization
- *Versatile with competitive results:*
classical planning, continuous planning, multiagent planning

Long-Term Vision: a self-motivated and self-aware dialogue agent

List of Publications & Articles Submitted for Publication

- Daphne Liu and Lenhart Schubert. Towards Self-Motivated, Cognitive, Continually Planning Agent. Manuscript submitted in February 2012 for publication (under review at *Computational Intelligence*).
- Daphne H. Liu and Lenhart Schubert. An Infrastructure for Self-Motivated, Continually Planning Agents in Virtual Worlds. *Technical Report 2012-985*, Dept. of Computer Science, University of Rochester, December 2012.
- Daphne Liu and L. K. Schubert. Combining Self-Motivation with Logical Planning and Inference in a Reward Seeking Agent. In *Proceedings of the International Conference on Agents and Artificial Intelligence, vol. 2*, January 2010.
- Daphne Liu and Lenhart Schubert. Incorporating Planning and Reasoning into a Self-Motivated, Communicative Agent. In *Proceedings of the Second Conference on Artificial General Intelligence*, March 2009.
- Daphne Hao Liu. A Survey of Planning in Intelligent Agents: from Externally Motivated to Internally Motivated Systems. *Technical Report 2008-936*, Dept. of Computer Science, University of Rochester, June 2008.