Human-in-the-Loop Planning & Decision Support

AAAI 2015 Tutorial

Slides at www.ktalamad.com/aaai-tutorial

(or google “rao asu”)

Subbarao Kambhampati
Arizona State University

Kartik Talamadupula
IBM T.J. Watson Research Center

Funding from ONR, ARO and NSF gratefully acknowledged

Introductions

- Subbarao Kambhampati
- Kartik Talamadupula

Recent Advances in AI Planning: A Unified View

AAAI-2000 Tutorial
(MA2)

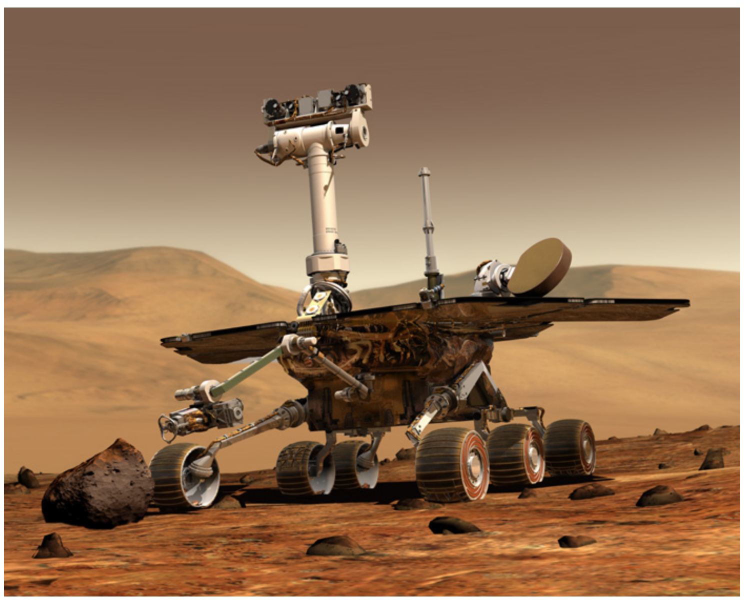
Subbarao Kambhampati
Arizona State University
rao@asu.edu

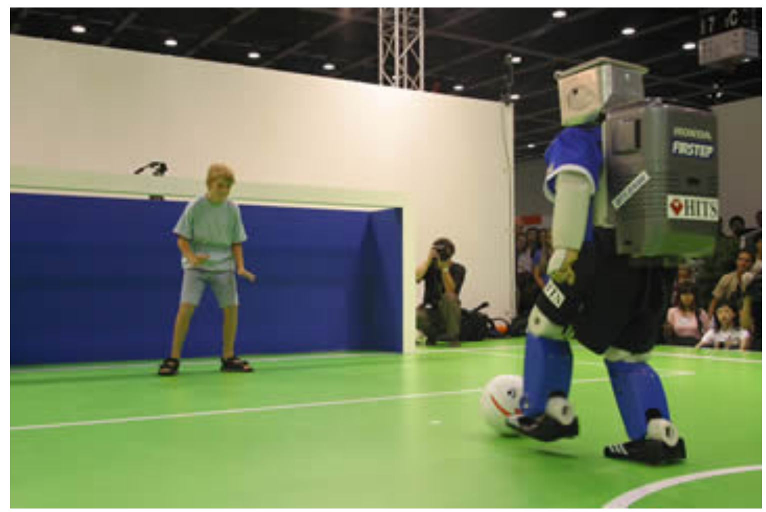
http://rakaposhi.eas.asu.edu/planning-tutorial

URL will have the up to date version

AI's Curious Ambivalence to humans..

- Our systems seem happiest
  - either far away from humans
  - or in an adversarial stance with humans





You want to help humanity, it is the people that you just can't stand…

What happened to Co-existence?

- Whither McCarthy's advice taker?
- ..or Janet Kolodner's house wife?
- …or even Dave's HAL?
  - (with hopefully a less sinister voice)

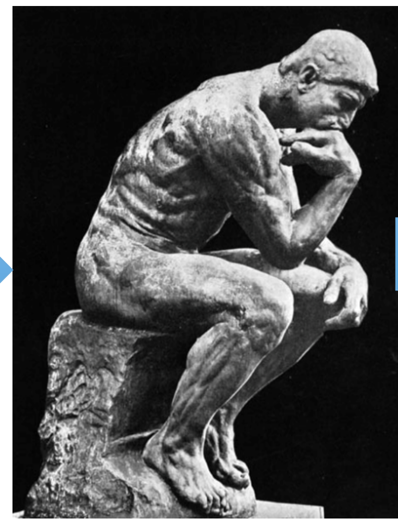
HAAI in Planning:
This Tutorial

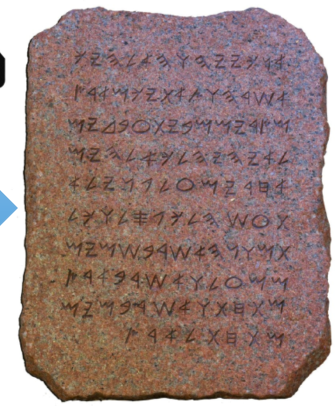
HAAI

Human-aware AI

Planning: The Canonical View

A fully specified
 problem
  --Initial state
  --Goals
      (each non-negotiable)
  --Complete Action Model





The Plan

Human-in-the-Loop Planning

- In many scenarios, humans are part of the planning loop, because the planner:
  - Needs to plan to avoid them
    - Human-Aware Planning
  - Needs to provide decision support to humans
      - Because “planning” in some scenarios is too important to be left to automated planners
    - “Mixed-initiative Planning”; “Human-Centered Planning”; “Crowd-Sourced Planning”
  - Needs help *from* humans
    - Mixed-initiative planning; “Symbiotic autonomy”
  - Needs to team with them
    - Human-robot teaming; Collaborative planning

Goal of this Tutorial

- Depending on the modality of interaction between human and the planner, HILP raises several open challenges for the planner
- The goal of this tutorial is to
  - Survey HILP scenarios
  - Discuss the dimensions along which they vary
  - Identify the planning challenges posed by HILP scenarios
    - Interpretation, Decision Support and Communication
  - Outline current approaches for addressing these challenges
  - Present detailed case-studies of two recent HILP systems (from our group ☺).

A Brief History of HILP

- Beginnings of significant interest in the 90's
  - Under the aegis of ARPA Planning Initiative (and NASA)
  - Several of the critical challenges were recognized
    - Trains project [Ferguson/Allen; Rochester]
    - Overview of challenges [Burstein/McDermott + ARPI Cohort]
    - MAPGEN work at NASA
  - At least some of the interest in HILP then was motivated by the need to use humans as a "crutch" to help the planner
    - Planners were very inefficient back then; and humans had to "enter the land of planners" and help their search..
- In the last ~15 years, much of the mainstream planning research has been geared towards improving the speed of of plan generation
  - Mostly using reachability and other heuristics; Helmert/Roger Tutorial this morning
- Renaissance of interest in HILP thanks to the realization that HILP is critical in many domains even with "fast" planners

BACK TO THE FUTURE of PLANNING

ACAI FREIBURG 2011

Lectures delivered at the ACAI Summer School on Automated Planning and Scheduling, June 2011

Abstract:

In its early days, the planning community routinely and gleefully let its reach exceed its grasp in terms of the class and scope of problems under consideration. Even when our planners were really classical but quite glacial, and could at best handle three blocks problems under mere minutes on a good day, we still blithely directed myriad efforts at lifted planning, temporal planning, stochastic planning, open world planning, mixed-initiative planning, and multi-agent planning.

The principled scale-up in classical planning in the last decade should have opened a more expansive vent for all that pent-up ambition. Alas, it hasn't quite turned out that way; our successes in scale-up seem to have turned us more circumspect. A Martian looking at any of the recent ICAPS proceedings can be forgiven for thinking that we are all mostly in quest of ever-more speed-up for classical planning.

In these lectures, I will make a case for turning our (and especially your) energies back to the future of planning, and explain how we can co-opt the scale-up in classical planning to aid in this quest. We shall look, in particular, towards advances in partial satisfaction planning, temporal planning, stochastic planning, as well as planning with incomplete models and open worlds.

Slides (final version; as delivered)

Audio Part 1

IMAGINE going BACK TO THE FUTURE of Planning

Imagine there's no Landmarks
It's easy if you try
No benchmarks below us
Above us only blai
Imagine all the planners
Planning for real

Imagine there's no state
It isn't hard to do
Nothing to regress or relax
And no cost guidance too
Imagine all the planners
Lifting all the worlds

You may say that I'm a whiner
But I'm not the only one
I hope someday you'll join us
And the ICAPS will be more fun

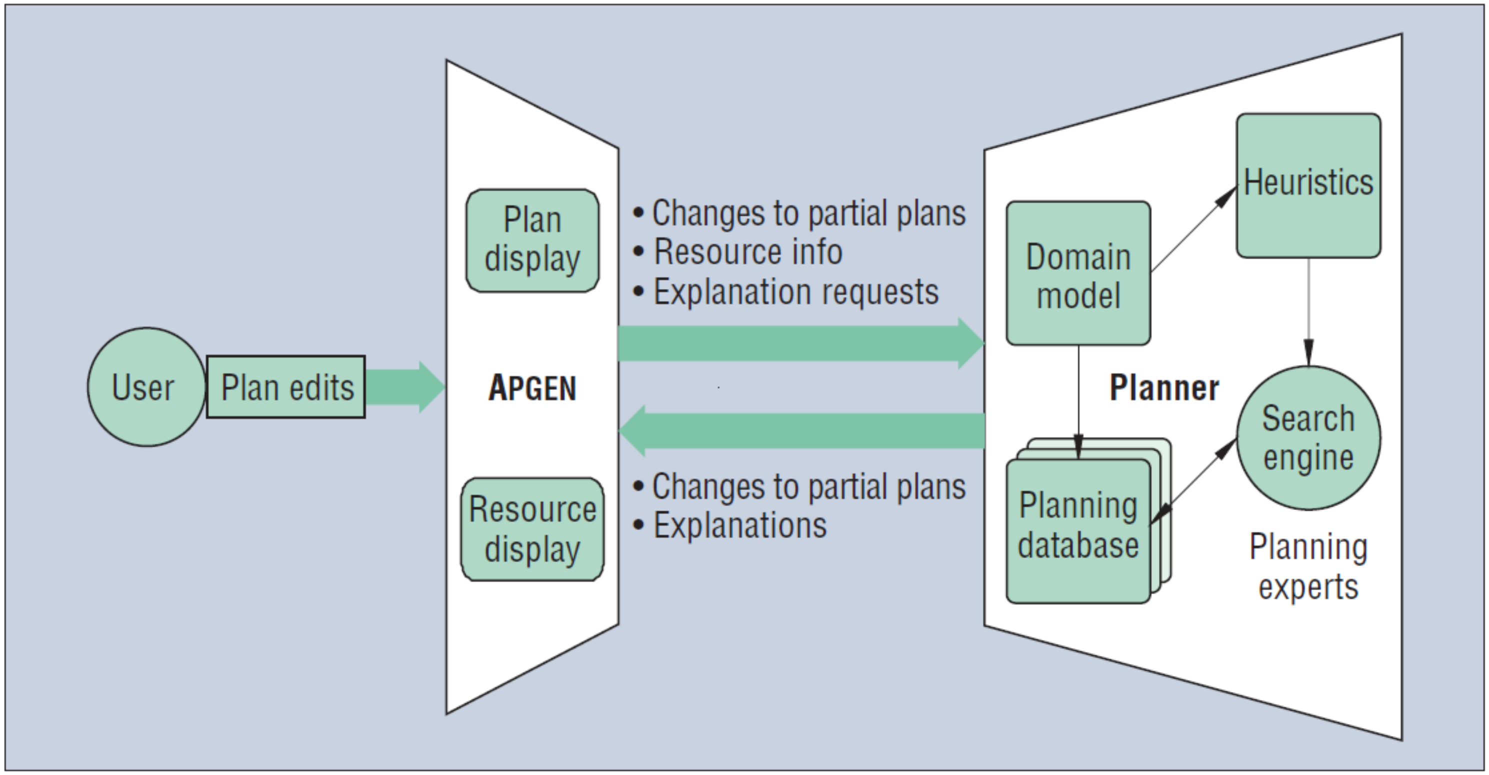
Imagine there's no models
I wonder if you can
No need for preferences or groundings
A diversity of plans
Imagine all the planners
Living life incomplete

You may say that I'm a whiner
But I'm not the only one
I hope someday you'll join us
And the ICAPS will be more fun

Quick Survey of some HILP Systems

(with a view to bring out the dimensions of variation)

MAPGEN—Mixed-initiative Planning



[Ai-Chang, Bresina et. Al., NASA;]

TRAINS

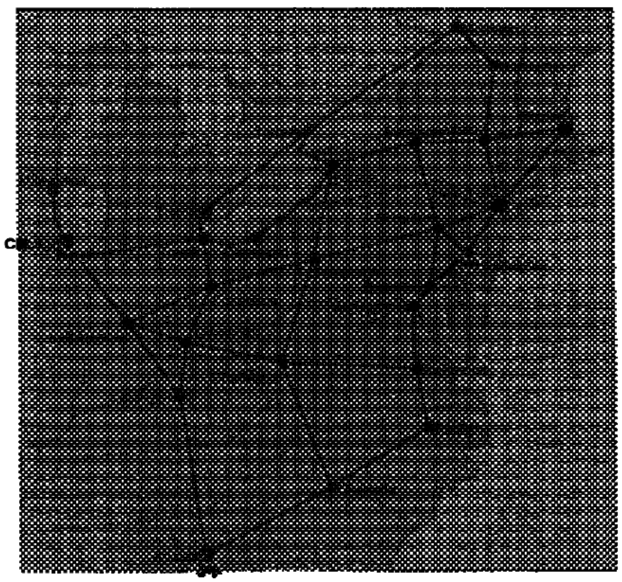


Figure 1: TRAINS-95 System Map Display

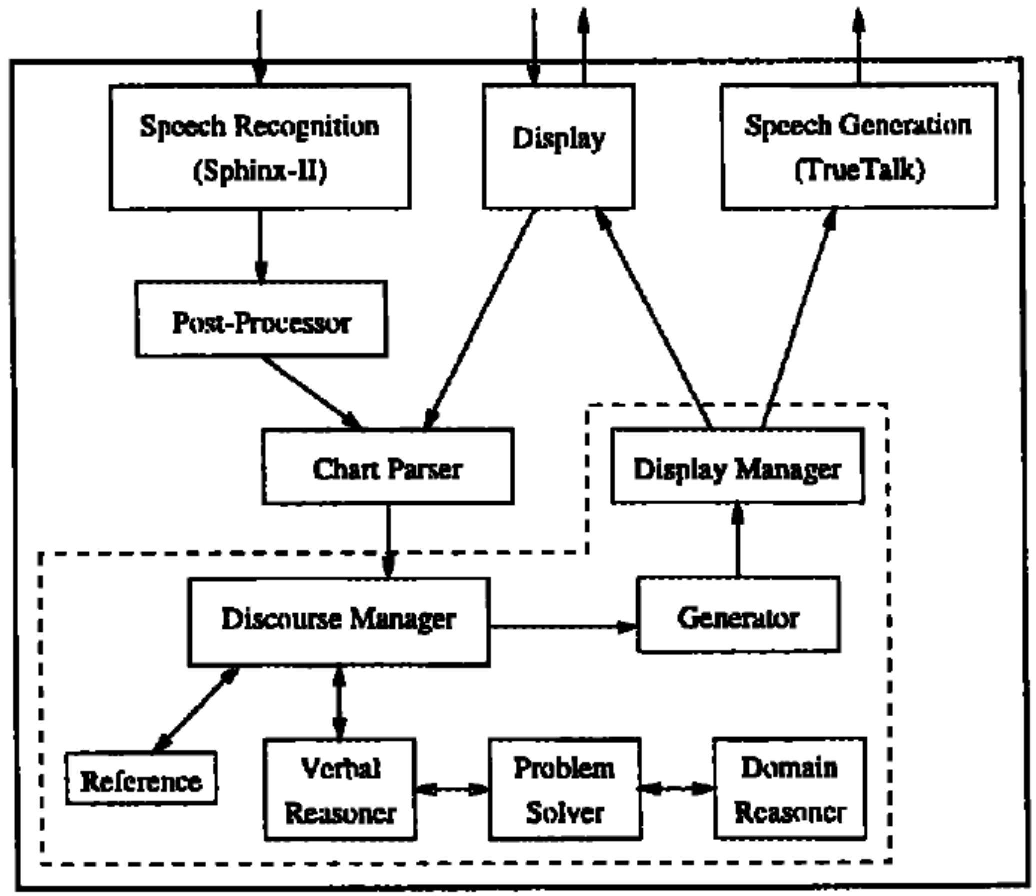


Figure 2: TRAINS-95 System Architecture

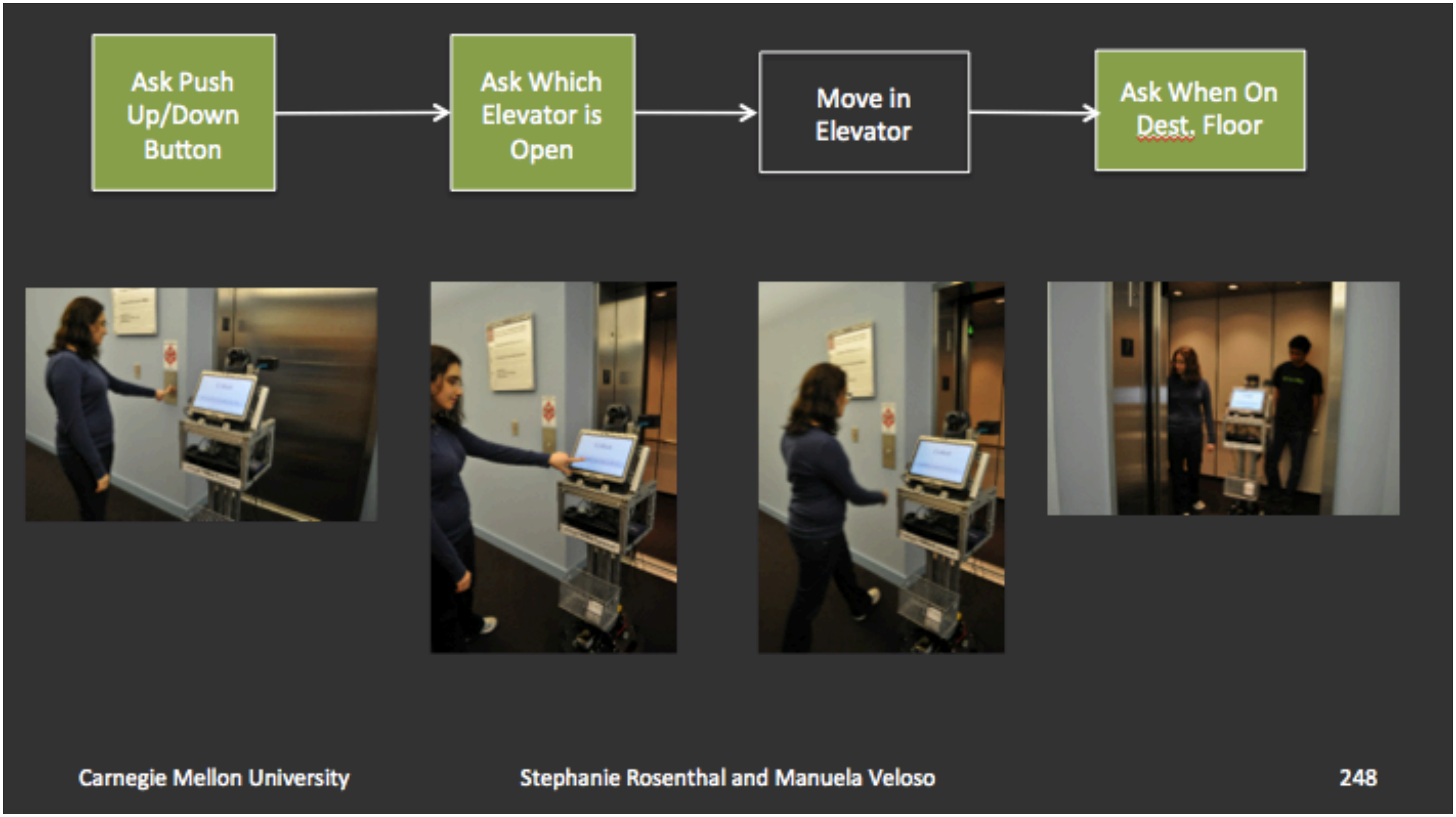
[Ferguson, Allen, Rochester; ~1995]

Motivation

- Task planning in inhabited environments
  (aka Human-aware Task Planning)
- Humans impose rules on acceptability of plans
  → "Grandpa hates robots"
  → "Don't vaccuum while I'm reading"
  → "Don't enter the bathroom when it's occupied"

Samuel Goldwyn Films, Robot & Frank (2012)

Planning to Use Human Help

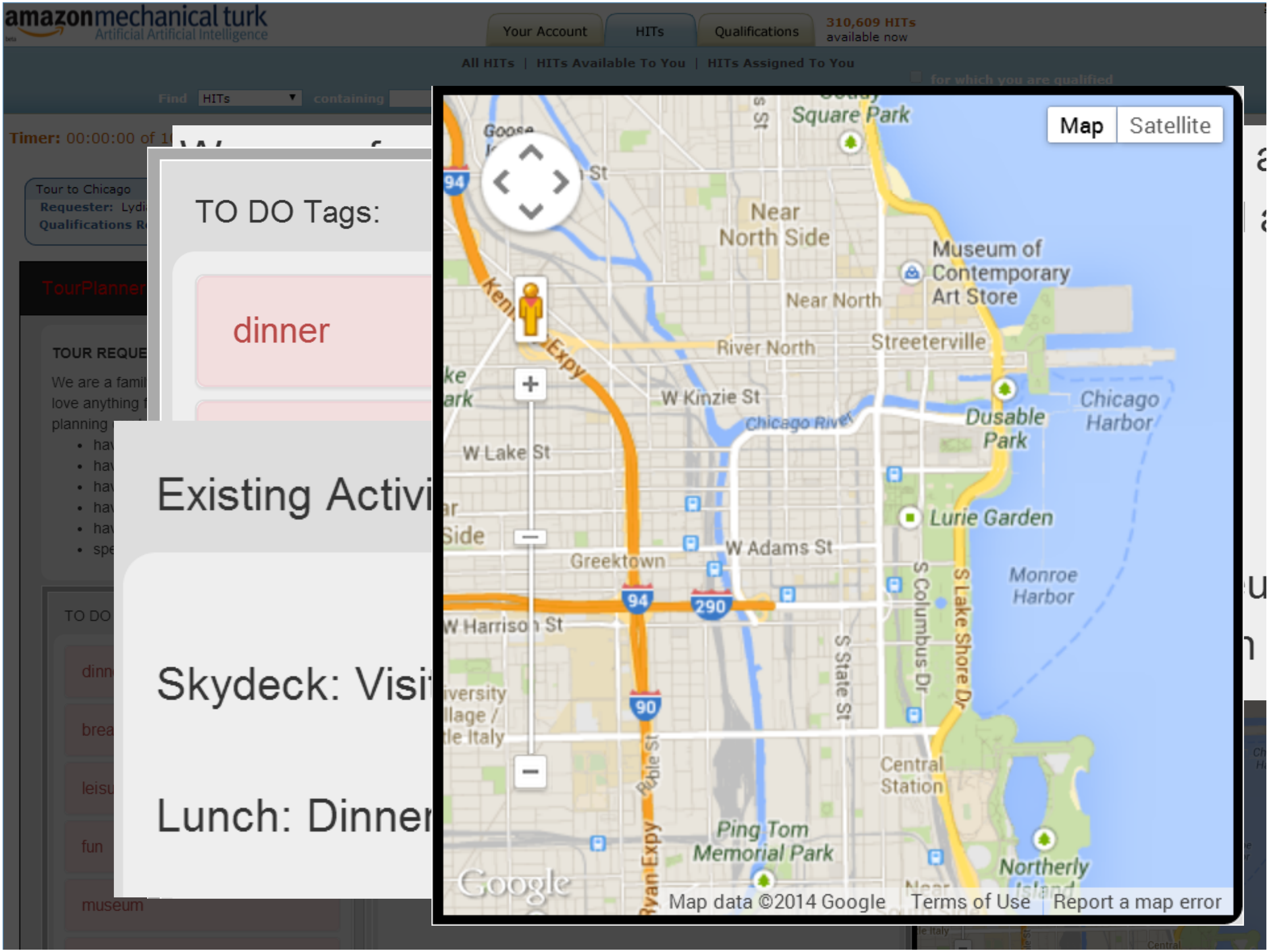


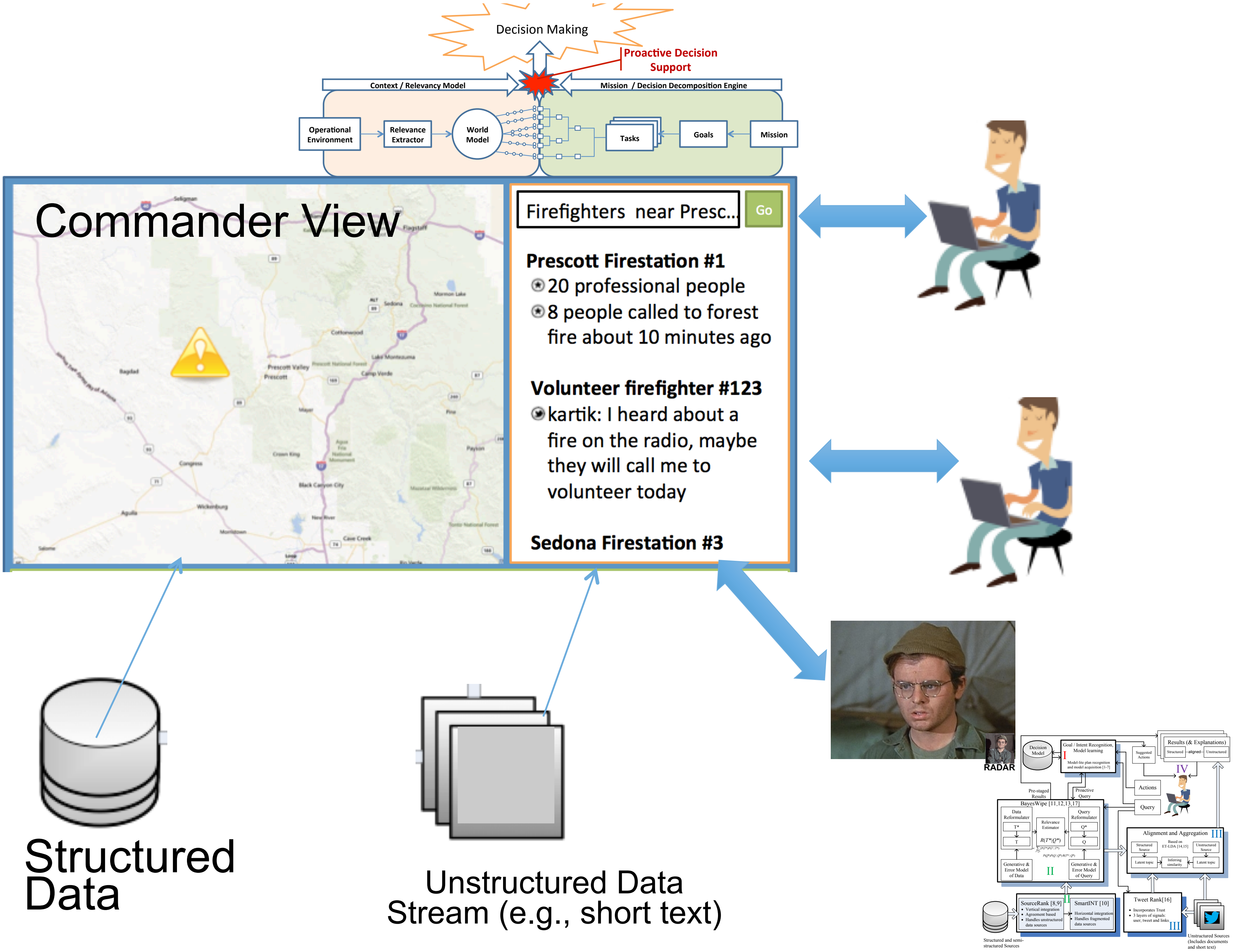
[Talamadupula, Scheutz, Kambhampati et. Al. 2010]

Human-Robot Teaming

- Search and report (rescue)
- Goals incoming on the go
- World is evolving
- Model is changing

- Infer instructions from Natural Language
- Determine goal formulation through clarifications and questions


amazonmechanical turk
Artificial Artificial Intelligence
Your Account
HITs
Qualifications
310,609 HITs
available now
All HITs | HITs Available To You | HITs Assigned To You
Find
HITs
containing
for which you are qualified
Timer: 00:00:00 of 1
Tour to Chicago
Requester: Lydi
Qualifications R
TourPlanner
TOUR REQUE
We are a famil
love anything f
planning
TO DO Tags:
dinner
Existing Activi
Skydeck: Visi
Lunch: Dinner
TO DO
dinn
brea
leisu
fun
museum
Map
Satellite
Square Park
Near North Side
Museum of Contemporary Art Store
Near North
River North
Streeterville
W Kinzie St
Chicago River
Dusable Park
Chicago Harbor
W Lake St
Lurie Garden
W Adams St
Greektown
Monroe Harbor
S Columbus Dr
S Lake Shore Dr
W Harrison St
S State St
Central Station
Ping Tom Memorial Park
Northerly Island
Google
Map data ©2014 Google
Terms of Use
Report a map error


Decision Making
Proactive Decision Support
Context / Relevancy Model
Mission / Decision Decomposition Engine
Operational Environment
Relevance Extractor
World Model
Tasks
Goals
Mission
Commander View
Firefighters near Presc...
Go
Prescott Firestation #1
20 professional people
8 people called to forest fire about 10 minutes ago
Volunteer firefighter #123
kartik: I heard about a fire on the radio, maybe they will call me to volunteer today
Sedona Firestation #3
RADAR
Structured Data
Unstructured Data Stream (e.g., short text)

Dimensions of Variation in Human in the Loop Planning

- Cooperation Modality
- Communication Modality
- What is Communicated
- Knowledge Level (Who knows what)

Cooperation Modality

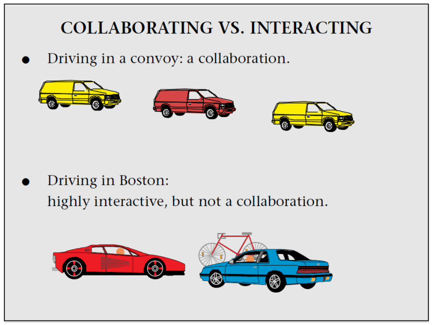


Figure 11. Collaborating versus Interacting.

- Awareness (No explicit communication)
  - Avoid getting into the human's way
    - Grandpa Hates Robots
  - Proactively support human's actions
- Interaction
  - Take commands/advice (Either via speech/language or via special interfaces)
  - From the human
    - Mapgen; Trains
  - From the Planner
    - CrowdPlanning; Radar
- Collaboration/Teaming
  - Human and Planner work together in formulating/executing the plan
    - Human-Robot Teaming

Communication Modality

- Through direct modification of plan structure
  - Mapgen
- Through custom interfaces
  - Quasi- stylized- natural language
    - Crowd Planning
- By speech and Natural Language
  - Trains, HRT
- Pre-specified constraints
  - Grandpa Hates Robots

What is communicated

- New goals
- New preferences & plan constraints
  - Grandpa Hates Robots: Interaction Constraints
  - Crowdplanning: Critiques, subgoals (from the planner)
- New model (actions etc)
  - Human Robot Teaming

Knowledge Level (Who knows What)

- Complete vs. Incomplete Models
  - Preference incompleteness (most of the time)
  - Dynamics incompleteness (sometimes)

Dimensions of HIL Planning

	Cooperation Modality	Communication Modality	What is Communicated	Knowledge Level
Crowdsourcing	Interaction (Advice from planner to humans)	Custom Interface	Critiques, subgoals	Incomplete Preferences Incomplete Dynamics
Human-Robot Teaming	Teaming/ Collaboration	Natural Language Speech	Goals, Tasks, Model information	Incomplete Preferences Incomplete Dynamics (Open World)
"Grandpa Hates Robots"	Awareness (pre-specified constraints)	Prespecified (Safety / Interaction Constraints)	No explicit communication	Incomplete Preferences Complete Dynamics
MAPGEN	Interaction (Planner takes binding advice from human)	Direct Modification of Plans	Direct modifications, decision alternatives	Incomplete Preferences Complete Dynamics

How do we adapt/ adopt the modern planning technology for HILP?

Planning: The Canonical View

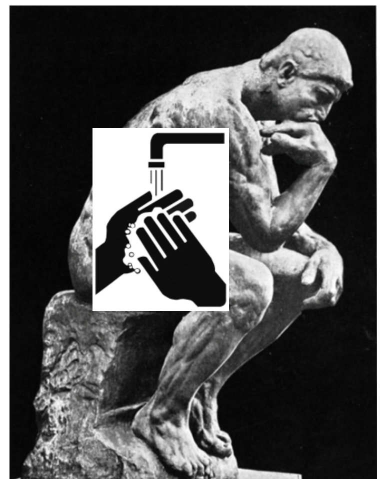
Full Problem Specification

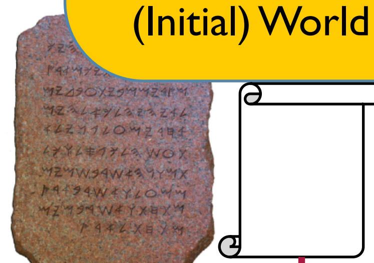
PLANNER

Fully Specified Action Model

Fully Specified Goals

Completely Known (Initial) World State





Plan (Handed off for Execution)

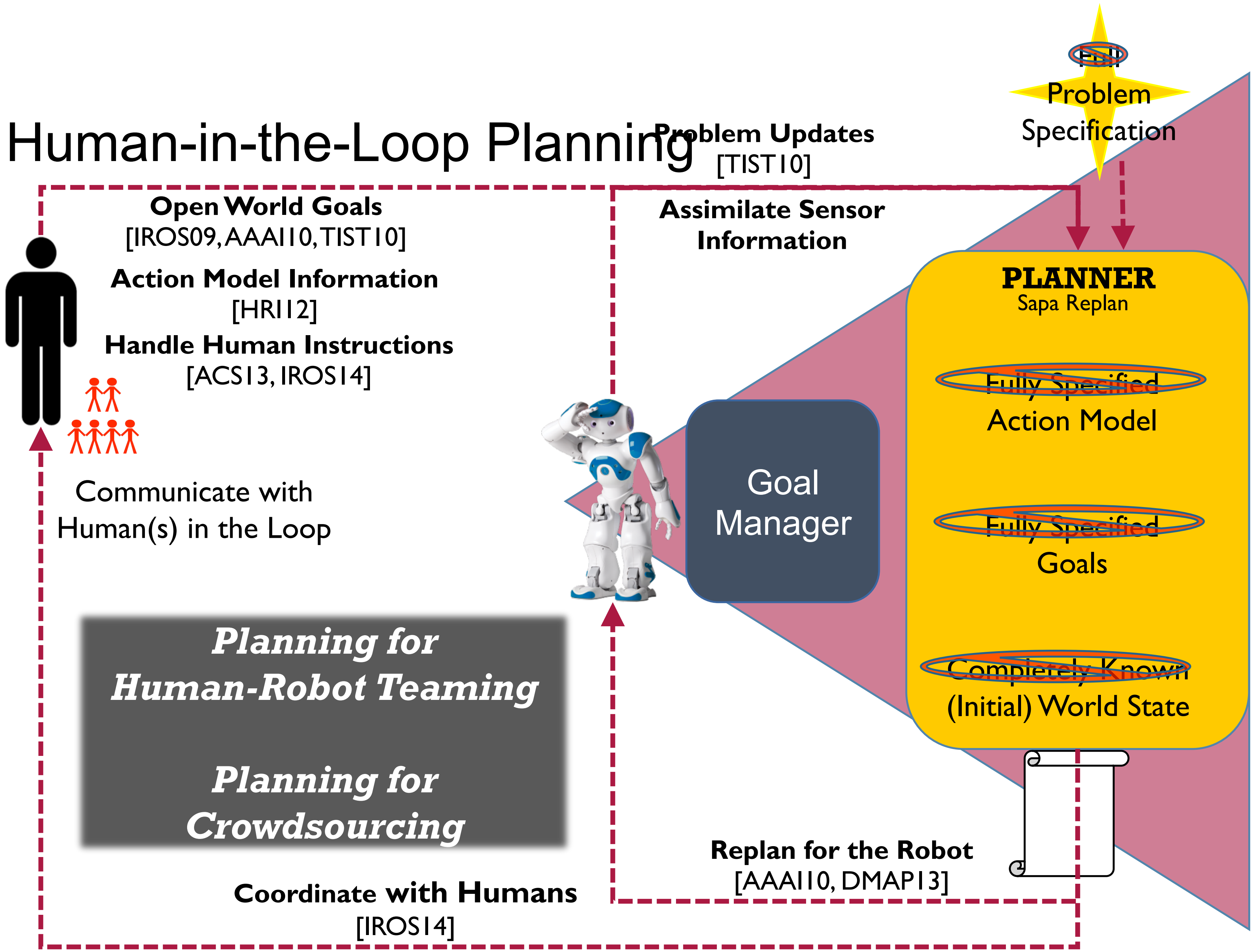
Violated Assumptions:

- → ~~Complete~~ Action Descriptions (Split knowledge)
- → ~~Fully Specified~~ Preferences (uncertain users)
- → ~~Packaged~~ planning problem (Plan Recognition)
- → ~~One-shot~~ planning (continual revision)

Planning is no longer a pure inference problem ☹

☹ But humans in the loop can ruin a really a perfect day ☹

Human-in-the-Loop Planning



Challenges for the Planner

- Interpret what humans are doing
  - Plan/goal/intent recognition
- Decision Support
  - Continual planning/Replanning
    - Commitment sensitive to ensure coherent interaction
    - Handle constraints on plan
  - Plan with incompleteness
    - Incomplete Preferences
    - Incomplete domain models
      - Robust planning with “lite” models
      - (Learn to improve domain models)
- Communication
  - Explanations/Excuses
    - Excuse generation can be modeled as the (conjugate of) planning problem
  - Asking for help/elaboration
    - Reason about the information value

(Other Relevant) Challenges (that are out-of-scope of this tutorial)

- Human Factors
  - How to make planning support “acceptable” to the humans in the loop?
  - How to adjust the planner autonomy to defer to the humans in the loop?
- Speech and Natural Language Processing in Collaborative Scenarios
- Learning to Improve models
  - Learning from demonstrations..
- Advances in multi-agent planning
  - Problem decomposition; Coordination etc.

OVERVIEW

1. INTRODUCTION [45]
2. INTERPRETATION [30]
3. DECISION SUPPORT [60]
   a. Explicit Constraints [30]
   b. Implicit Constraints (Preferences) [15]
   c. Incomplete Dynamics [15]
4. COMMUNICATION [30]
   a. Excuses & Explanations [15]
   b. Asking for Help [15]
5. CASE STUDY [30]
6. SUMMARY [15]

OVERVIEW

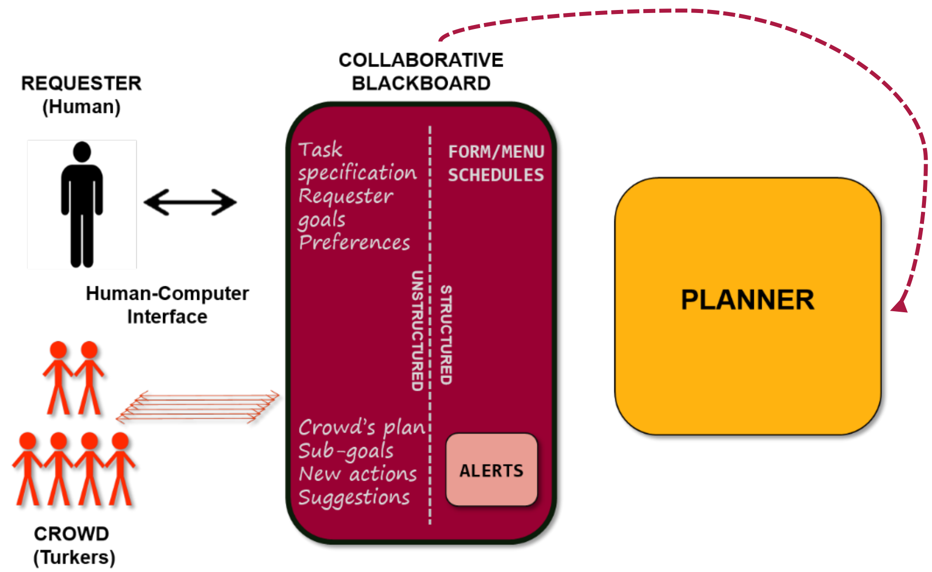
1. INTRODUCTION [45]
2. INTERPRETATION [30]
3. DECISION SUPPORT [60]
   a. Explicit Constraints [30]
   b. Implicit Constraints (Preferences) [15]
   c. Incomplete Dynamics [15]
4. COMMUNICATION [30]
   a. Excuses & Explanations [15]
   b. Asking for Help [15]
5. CASE STUDY [30]
6. SUMMARY [15]

OVERVIEW

1. INTRODUCTION
2. INTERPRETATION
3. DECISION SUPPORT
   a. Explicit Constraints
   b. Implicit Constraints (Preferences)
   c. Incomplete Dynamics
4. COMMUNICATION
   a. Excuses & Explanations
   b. Asking for Help
5. CASE STUDY
6. SUMMARY

CHALLENGE: INTERPRETATION

- Understanding the goals and plans of humans from semi-structured or unstructured text
- Impedance Mismatch



Extract from Plain Text
Impose structure
[Ling & Weld, 2010]
[Kim, Chacha & Shah, 2013]

Full Plan Recognition
[Kautz & Allen, 1986]
[Ramirez & Geffner, 2010]

UNSTRUCTURED **STRUCTURED**

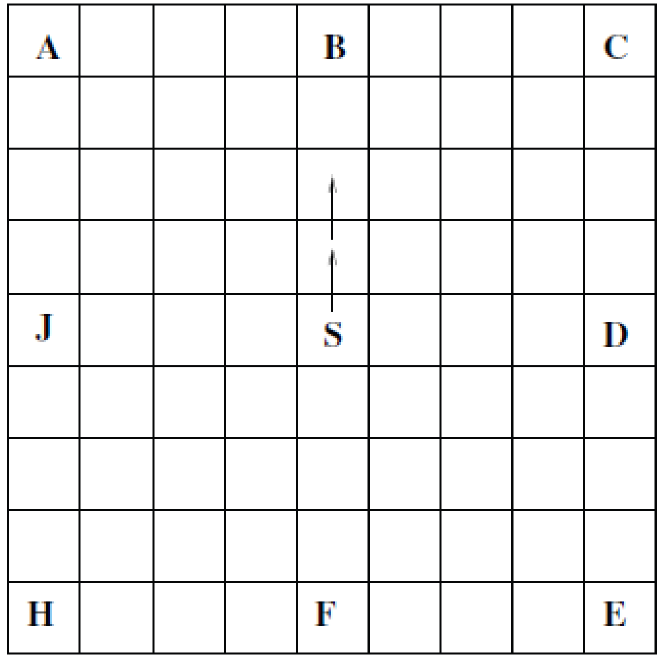
Plan Recognition from Noisy Traces
Extract noisy traces first
[Zhuo, Yang & Kambhampati, 2012]

Dealing with Interpretation

- **Assume Structure**
  - Exploit/assume structured representation (plan)
  - Easier to match planner's expectation of structured input
  - Restricts flexibility of humans; less knowledge specified

- **Extract/Infer Structure**
  - Allow humans to use natural language
    - Semi-structured and unstructured text
  - Extract information from human-generated input
  - Validate against (partial) model
  - Iteratively refine recognized goals and plan

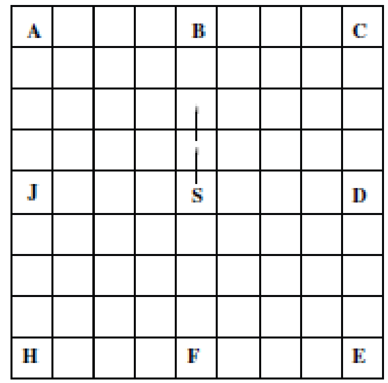
Plan Recognition



- Agent can **move** one unit in the four directions
- Possible **targets** are A, B, C, . . .
- Starting in S, he is **observed** to move up twice
- **Where** is he going? Why?

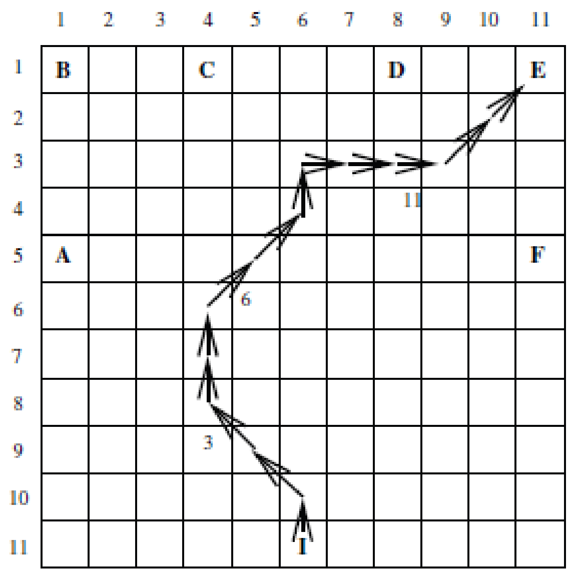
Plan Recognition as Planning
Miquel Ramirez
Hector Geffner

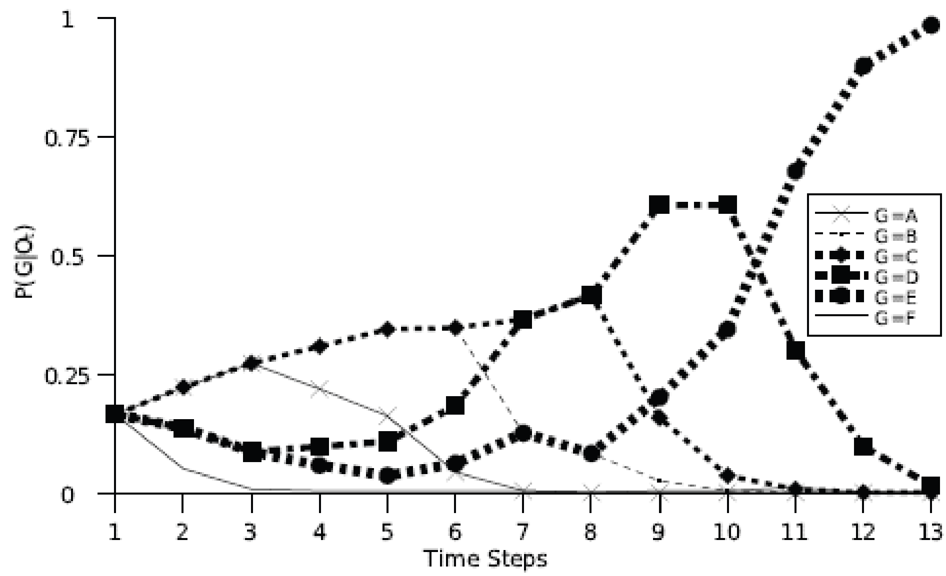
Example (cont'd)



- From Bayes, **goal posterior** is $P(G|O) = \alpha\, P(O|G)\, P(G)$, $G \in \mathcal{G}$
- If **priors** $P(G)$ given for each goal in $\mathcal{G}$, the question is what is $P(O|G)$?
- $P(O|G)$ measures **how well goal** G **predicts observed actions** O
- In **classical** setting,
  - ▷ G predicts O **best** when need to get off the way **not** to comply with O
  - ▷ G predicts O **worst** when need to get off the way **to comply with** O

Illustration: Noisy Walk





Graph on left shows 'noisy walk' and possible targets; curves on right show resulting **posterior probabilities** $P(G|O)$ of each possible target G as a function of time

Approach to plan recognition can be generalized to other models (MDPs, POMDPs); the idea is that if you have a **planner** for a model, then you also have a **plan recognizer** for that model given a **pool of possible goals.**

Beliefs, Intentions & Teaming

- Agents have **beliefs** and **intentions**
  - An agent can model its *team members'* beliefs and intentions

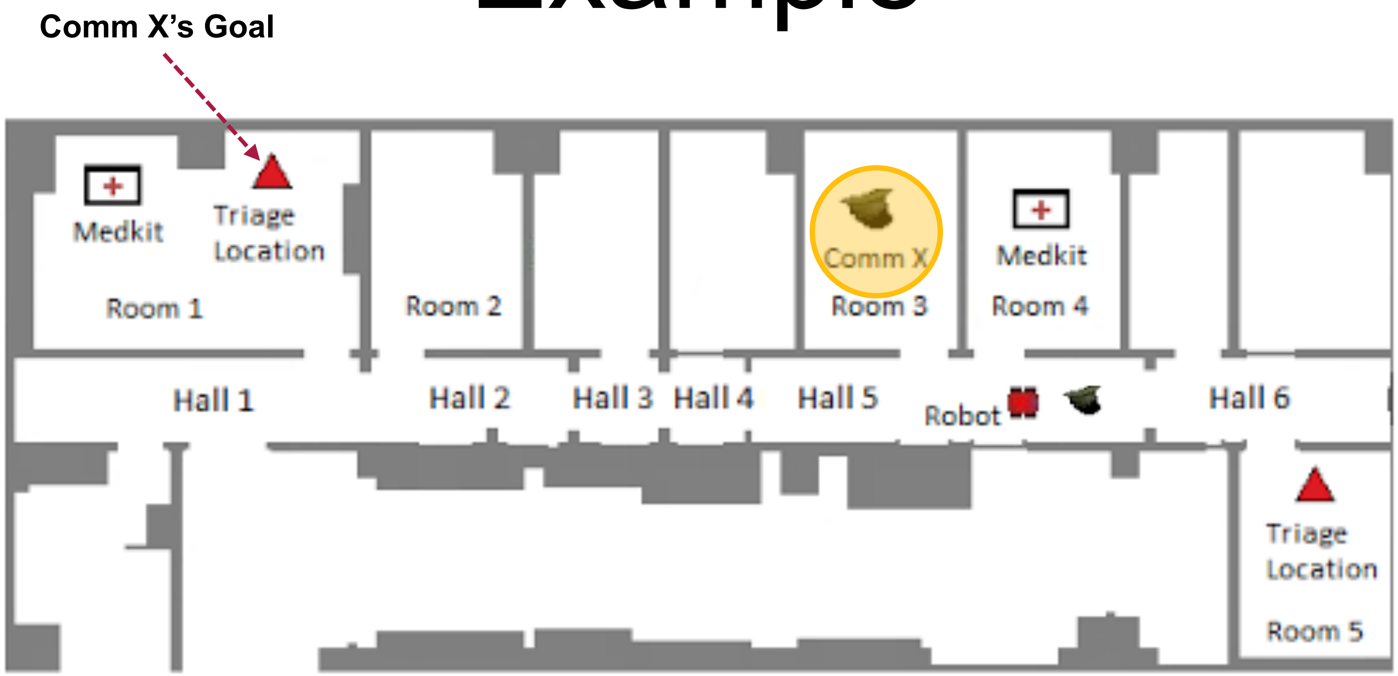
$$\{\ \phi \mid bel(\alpha, \phi)\ \in\ Bel_{self}\ \}$$
$$\{goal(\alpha, \phi, P) | goal(\alpha, \phi, P) \in Bel_{self}\}$$

- This information can be used to **predict the plans** of team members

Automated Planning

- Used for high-level plan synthesis
- Can be used to **simulate** an agent's plan
  - Based on known beliefs and intentions
  - Some information about agent's capabilities
- Automated Planning Instance:
  - **Initial State**: All known beliefs of that agent
  - **Goal Formula**: All known goals of that agent
  - **Action Model**: Precondition/Effect description

Example



PREDICTED PLAN

move commx room3 hall5
move_reverse commx hall5 hall4
move_reverse commx hall4 hall3
move_reverse commx hall3 hall2
move_reverse commx hall2 hall1
move_reverse commx hall1 room1
pick_up_medkit commx mkeast room1
conduct_triage commx room1

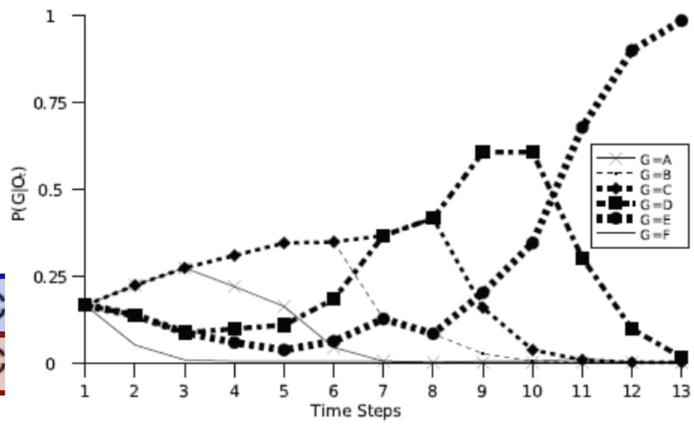
But what if we don't have full knowledge regarding the team member's goal(s)?

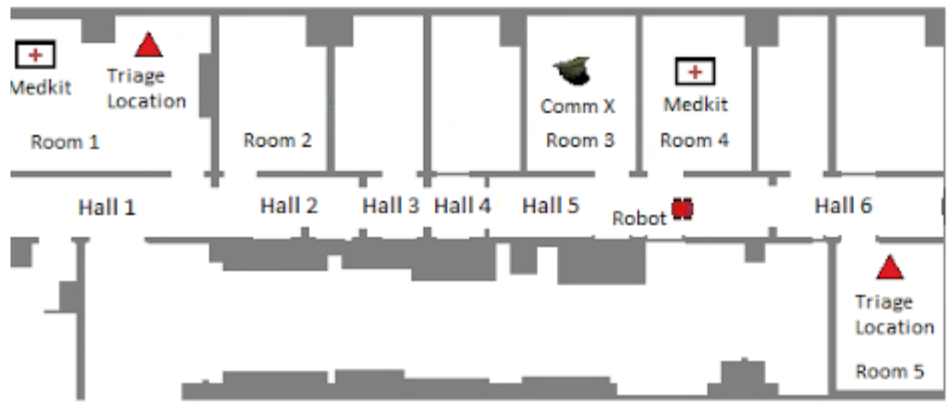
Plan Recognition

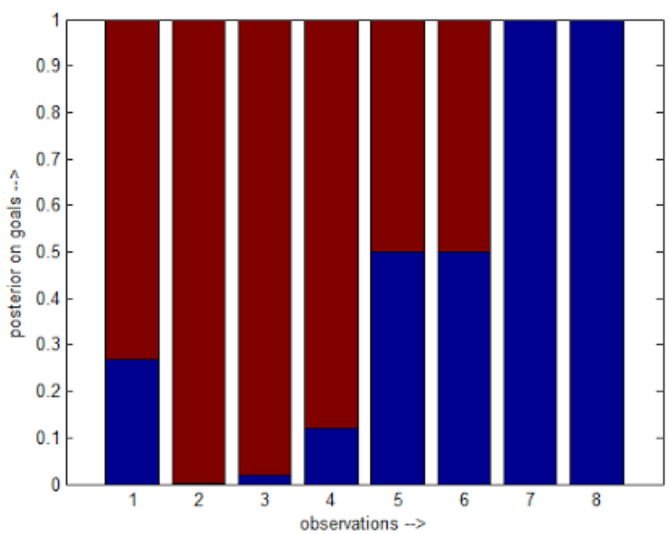
BELIEF

(conduc

(conduc

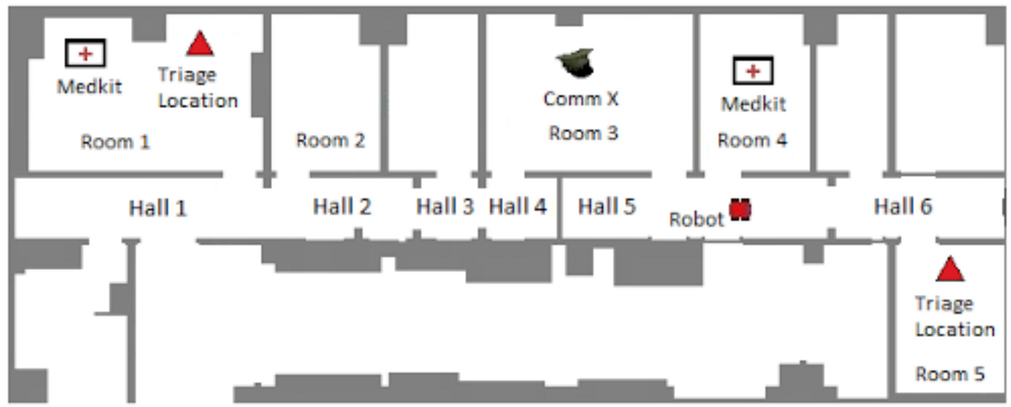






observations -
move commx room3 hall5
move_reverse commx hall5 hall4
move_reverse commx hall4 hall3
move_reverse commx hall3 hall2
move_reverse commx hall2 hall1
move_reverse commx hall1 room1
pick_up_medkit commx mkeast room1
conduct_triage commx room1

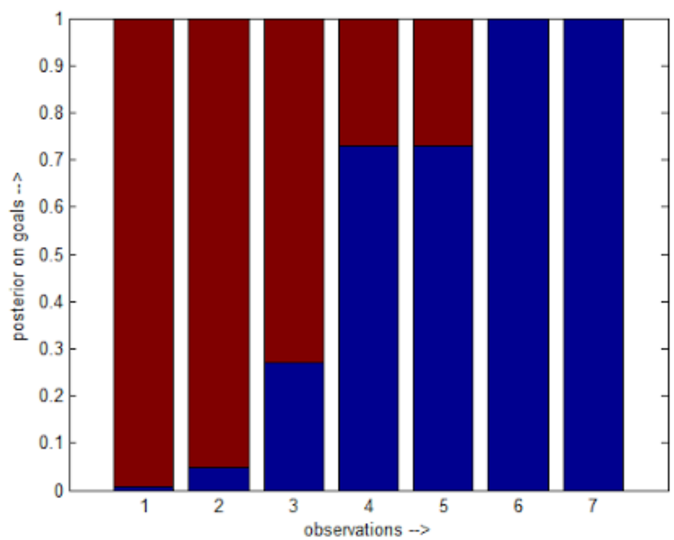
Plan Recognition



BELIEF IN GOAL

(conducted_triage commX room1)

(conducted_triage commX room5)



observations -
move commx room3 hall4
move_reverse commx hall4 hall3
move_reverse commx hall3 hall2
move_reverse commx hall2 hall1
move_reverse commx hall1 room1
pick_up_medkit commx mkeast room1
conduct_triage commx room1

DEALING WITH INTERPRETATION

- **Assume Structure**
  - Exploit/assume structured representation (plan)
  - Easier to match planner's expectation of structured input
  - Restricts flexibility of humans; less knowledge specified

- **Extract/Infer Structure**
  - Allow humans to use natural language
    - Semi-structured and unstructured text
  - Extract information from human-generated input
  - Validate against (partial) model
  - Iteratively refine recognized goals and plan

Why Infer Task Plans?

- Integrate robots seamlessly in time-critical domains
- Lessen burden of programming and deploying robots
- Leverage the use of web-based planning tool (NICS)

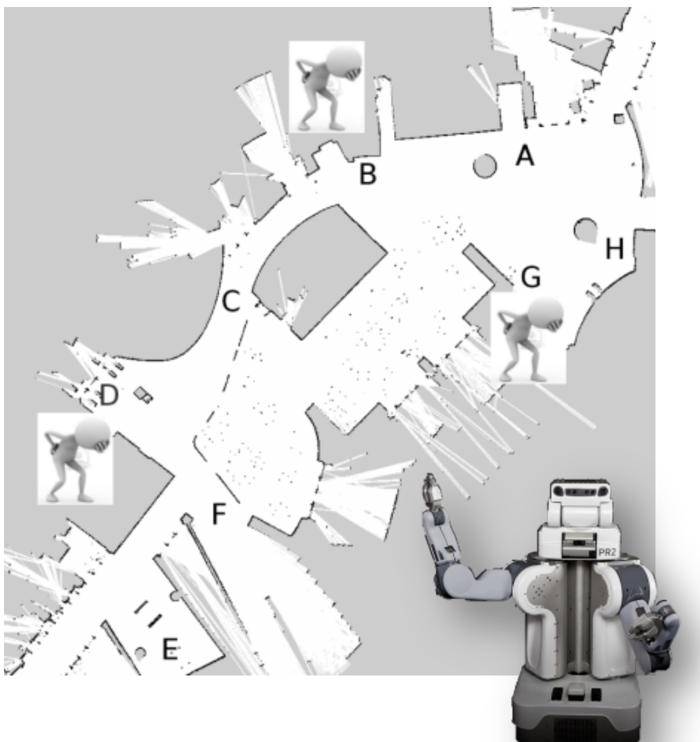
Planning Conversation

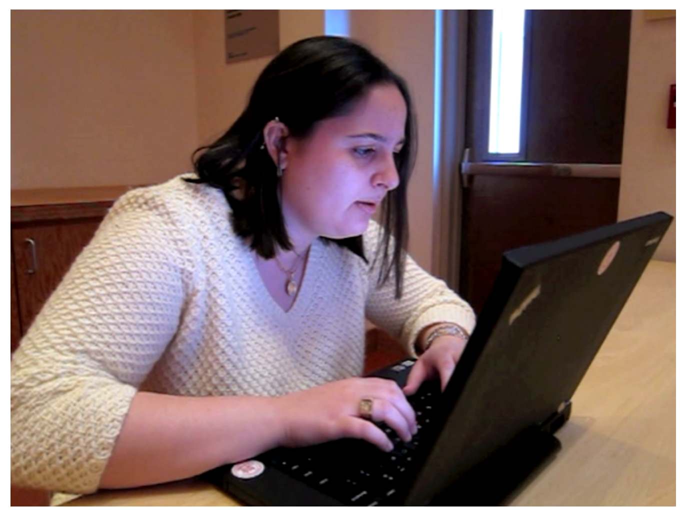
Robot Task Plans

Inferring Robot Task Plans from Human Team Meetings

Been Kim, Caleb Chacha and Prof. Julie Shah
MIT

Human Team Planning





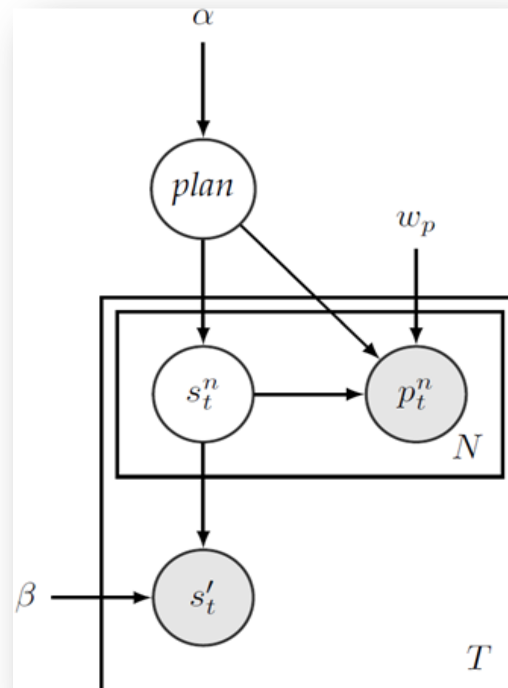
General Framework

Raw Planning Conversation Data from Web-based Tool

Algorithm Input:
Structured Form of Noisy Planning Data

U1: Send(blue robot, B)
while Send(red robot, G)
Then Send(medical, B)…
…….
U9: Send(mechanics, C)

Algorithm:
Sampling Inference in Generative Model + Logic-based prior (PDDL)



Algorithm Output:
Final Agreed Plan

1. Send Red Robot to B
2. Send Blue Robot to A
3. Send Red Medical Crew G
…..

Robot Task Plans:

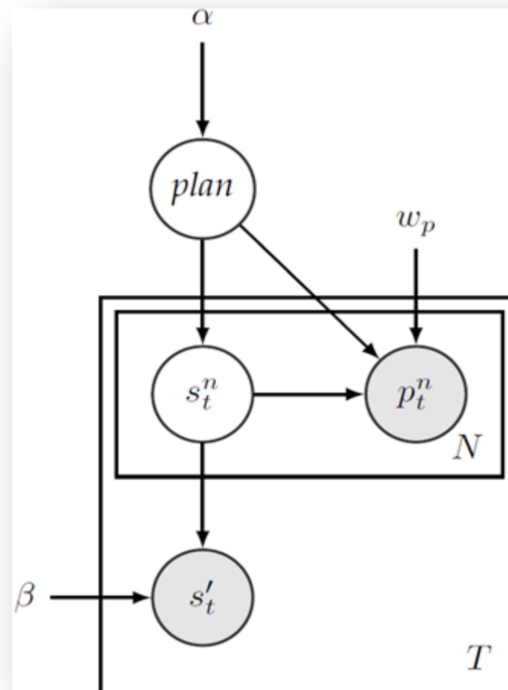
Algorithm Input

Raw Planning Conversation Data from Web-based Tool

Algorithm Input:
Structured Form of Noisy Planning Data

U1: Send(blue robot, B)
while Send(red robot, G)
Then Send(medical, B)…
…….
U9: Send(mechanics, C)

Algorithm:
Sampling Inference in Generative Model + Logic-based prior (PDDL)



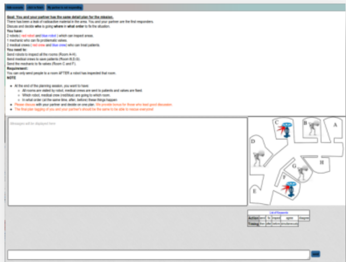
Algorithm Output:
Final Agreed Plan

1. Send Red Robot to B
2. Send Blue Robot to A
3. Send Red Medical Crew G

…..

Robot Task Plans:

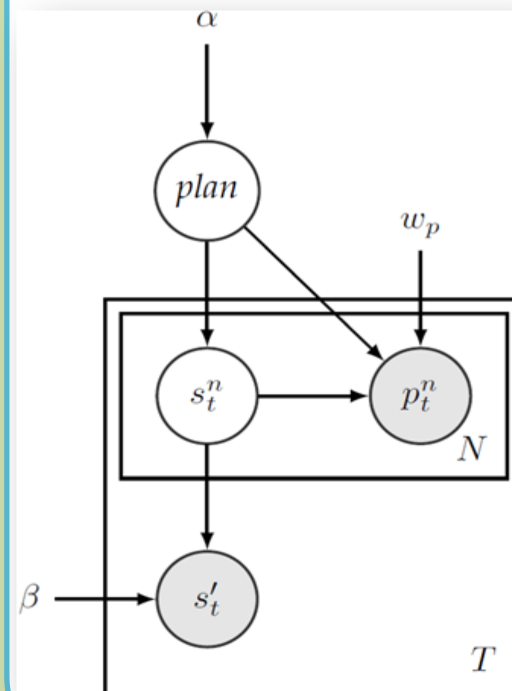
Raw Planning Conversation Data from Web-based Tool



Algorithm Input:
Structured Form of Noisy Planning Data

U1: Send(blue robot, B)
while Send(red robot, G)
Then Send(medical, B)…
…….
U9: Send(mechanics, C)

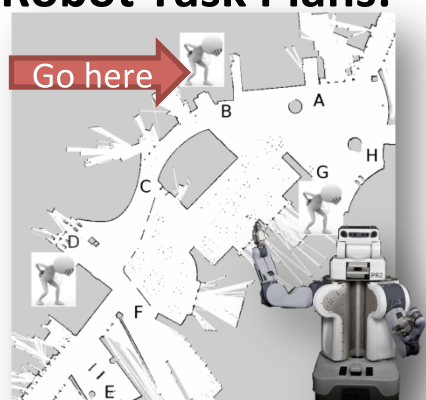
Algorithm:
Sampling Inference in Generative Model
+ Logic-based prior (PDDL)



Algorithm Output:
Final Agreed Plan

1. Send Red Robot to B
2. Send Blue Robot to A
3. Send Red Medical Crew G
…..

Robot Task Plans:

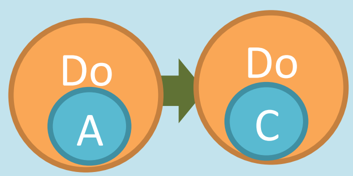


Input and Output

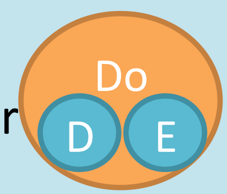
Input: structured planning conversation

Relatively ordered noisy plan parts

What we see:
Planning Conversation



Me: First, let's do A then C?
({A}, {C})



You: We should do D and E together
({D,E})

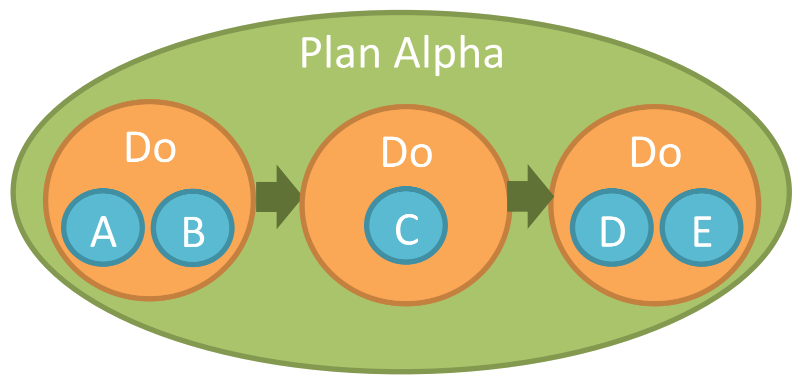
Me: Great, we can do H after that.
({H})

Output: Final plan

Absolutely orders sets of actions

({A,B}, {C}, {D,E})

Ordered tuple of sets of grounded predicate[1]



[1] Howey, R.; Long, D.; and Fox, M. 2004. Val: Automatic plan validation, continuous effects and mixed initiative planning using PDDL. In ICTAI

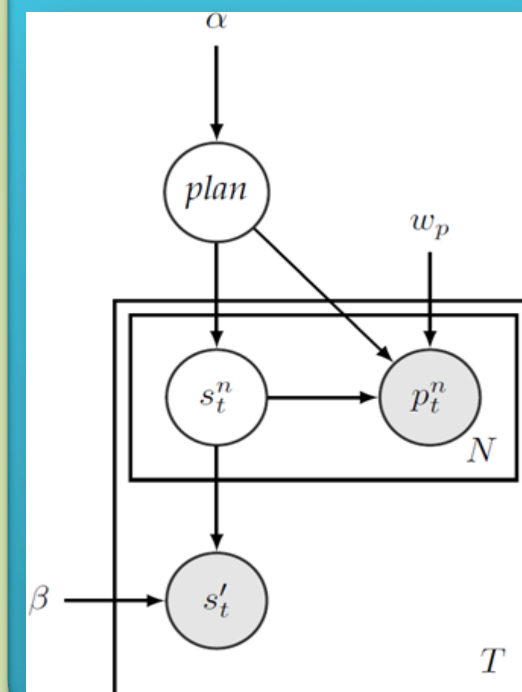
Algorithm

Raw Planning Conversation Data from Web-based Tool

Algorithm Input:
Structured Form of Noisy Planning Data

U1: Send(blue robot, B)
while Send(red robot, G)
Then Send(medical, B)…
…….
U9: Send(mechanics, C)

Algorithm:
Sampling Inference in Generative Model
+ Logic-based prior (PDDL)



Algorithm Output:
Final Agreed Plan

1. Send Red Robot to B
2. Send Blue Robot to A
3. Send Red Medical Crew G
…..

Robot Task Plans:

Approach

Logical Approach

- Can be solved as a logical constraint problem of partial order planning
- Fails in processing noisy data

Probabilistic Approach

- Small data (succinct conversation)
- Large solution space

→ Uninformative prior will have to search through a big space

Probabilistic Generative Modeling with Logic Based Prior

- Logic based plan validator (PDDL plan validator) to build informative prior
- Can deal with noisy data

What is PDDL Validator?

PDDL Validator Input

Domain Definition File

- Available actions
- Pre/post conditions for actions

Problem Definition File

- Available resources
- Initial condition
- Goal condition

Candidate Plan File

PDDL Plan Validator

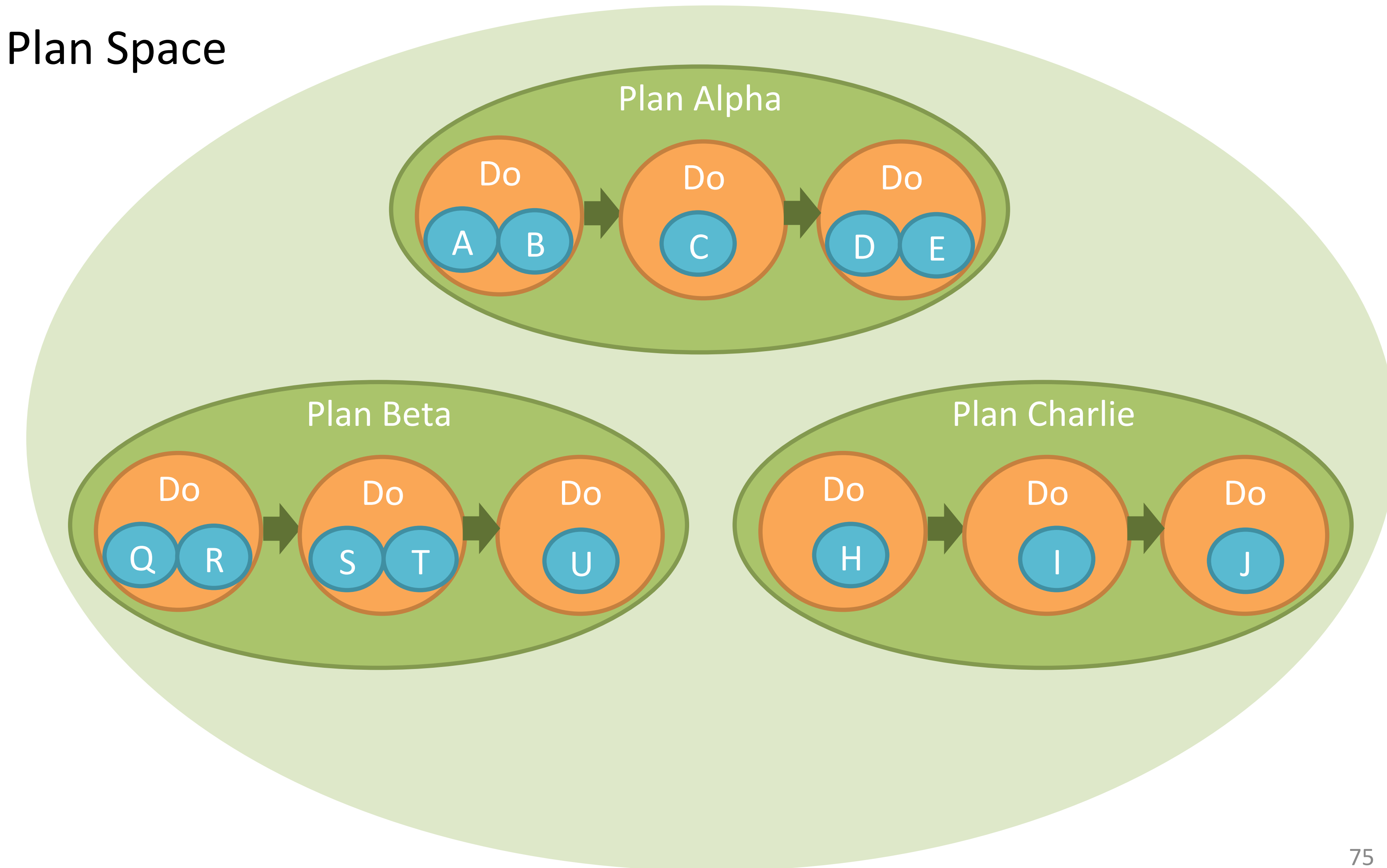
PDDL Validator Output

Yes, it is a valid plan

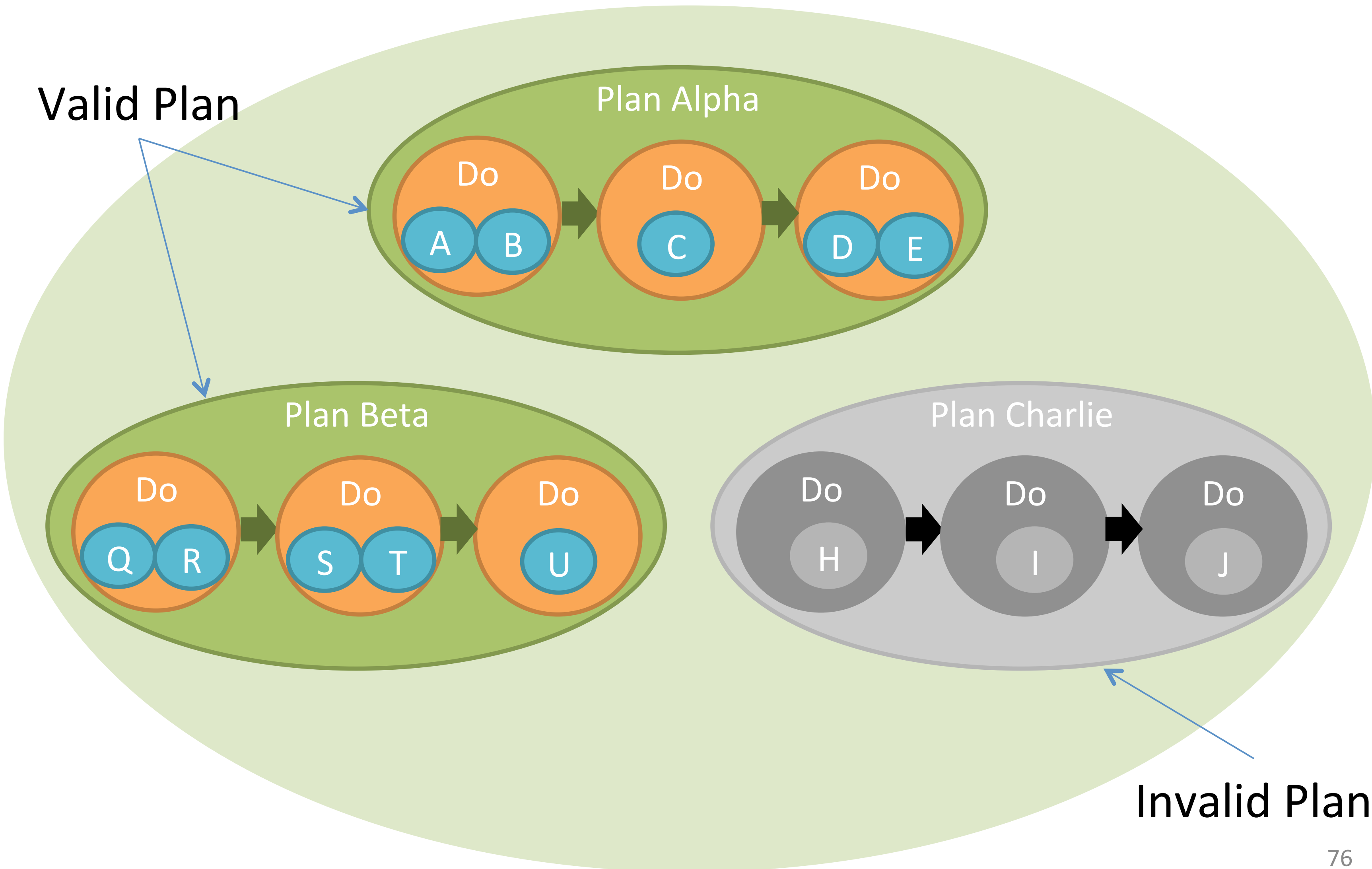
No, it is not a valid plan

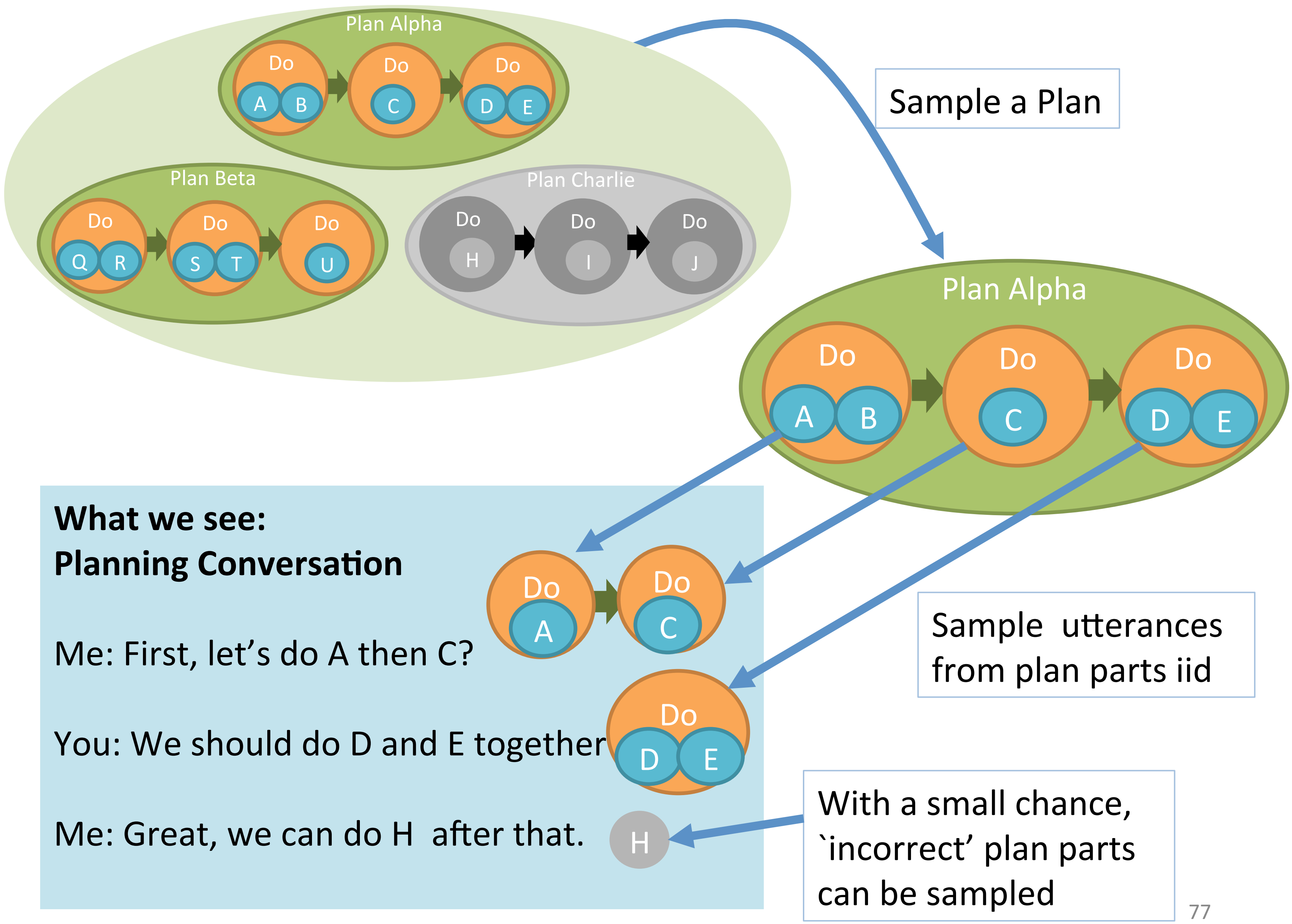
The algorithm is also tested with imperfect input files

Generative Model

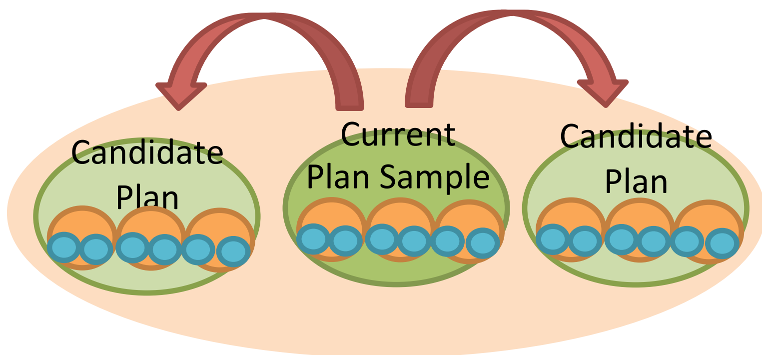


Generative Model




Plan Alpha
Do A B
Do C
Do D E
Plan Beta
Do Q R
Do S T
Do U
Plan Charlie
Do H
Do I
Do J
Sample a Plan
Plan Alpha
Do A B
Do C
Do D E
What we see:
Planning Conversation
Me: First, let’s do A then C?
Do A
Do C
You: We should do D and E together
Do D E
Me: Great, we can do H after that.
H
Sample utterances from plan parts iid
With a small chance, `incorrect’ plan parts can be sampled

PDDL Validator in Gibbs Sampling



Neighboring Plans

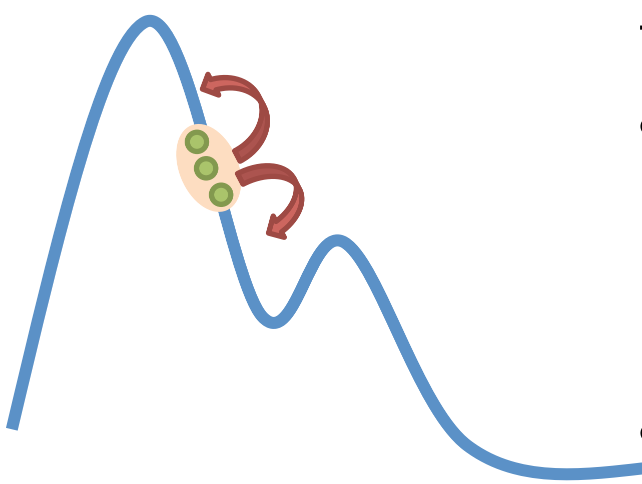
- Problem with sampling plan
  - No obvious conjugate prior
  - Intractable to calculate normalization constant

→ Metropolis-Hastings sampling

- PDDL plan validator is used to score the candidate plan

$$P(\textit{candidate plan} \mid \textit{everything else})$$

- Intuitively, maps what humans are good at for machines



Posterior distribution

Summary

- Logical plan validator + Probabilistic generative model to perform efficient inference
- Inferring the robot's plan in context with the full joint plan

Next Steps

- Include the ordering of conversation in the model – Large scale complex planning data

Structured Interpretation

For Multi-Agent Systems

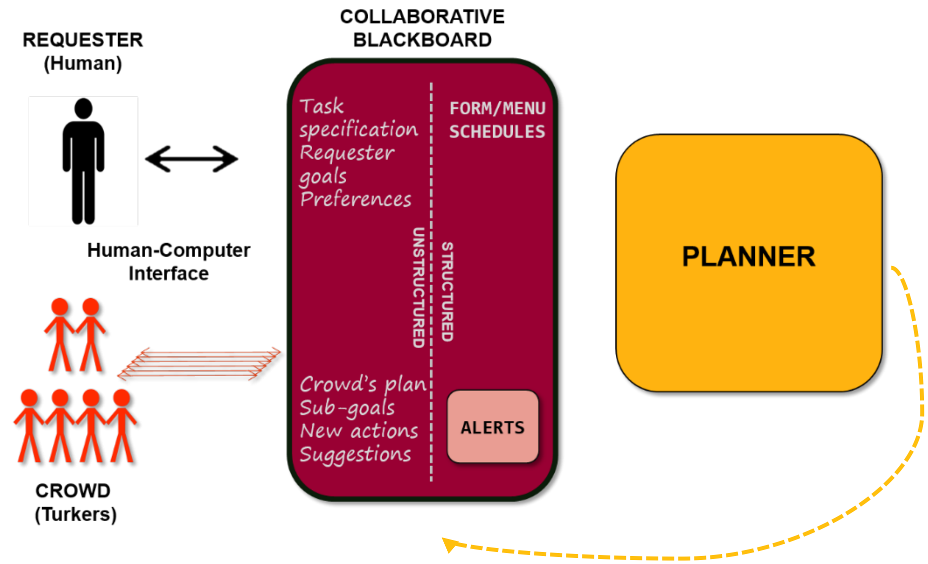
Systems	Inputs			
	Team trace	Action models	Plan graph	Plan cases
Banerjee et al. AAAI2010	full	×	×	full
Banerjee et al. AAAI2011	full	×	full	×
Zhuo et al. IJCAI2011	partial	×	×	partial
DARE Zhuo et al. NIPS 2012	partial	full	×	×

OVERVIEW

1. INTRODUCTION
2. INTERPRETATION
3. DECISION SUPPORT
   a. Explicit Constraints
   b. Implicit Constraints (Preferences)
   c. Incomplete Dynamics
4. COMMUNICATION
   a. Excuses & Explanations
   b. Asking for Help
5. CASE STUDY
6. SUMMARY

CHALLENGE: DECISION SUPPORT

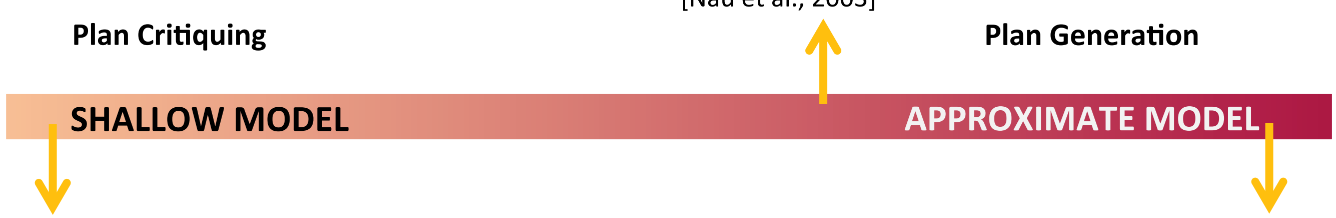
- Steering the human(s) to help in producing & critiquing a plan
  - Partial domain dynamics
  - Incomplete preferences
- Iterative Process



Problem Decomposition
[Nau et al., 2003]

Plan Critiquing

Plan Generation



Constraint & Arithmetic Checking
Mobi [Zhang et al., 2012]
Cobi [Zhang et al., 2013]

Regression
Match sub-goals
with actions

Continual Planning

Decision Support

- **New Information**
  - Sources: Sensors, Other Agents
- **New Goals**
  - From humans: Orders
  - From other agents: Requests
- **Commitments**
  - Publication of currently executing plan creates *commitments* in the world
    - Other agents may base their plans on this plan

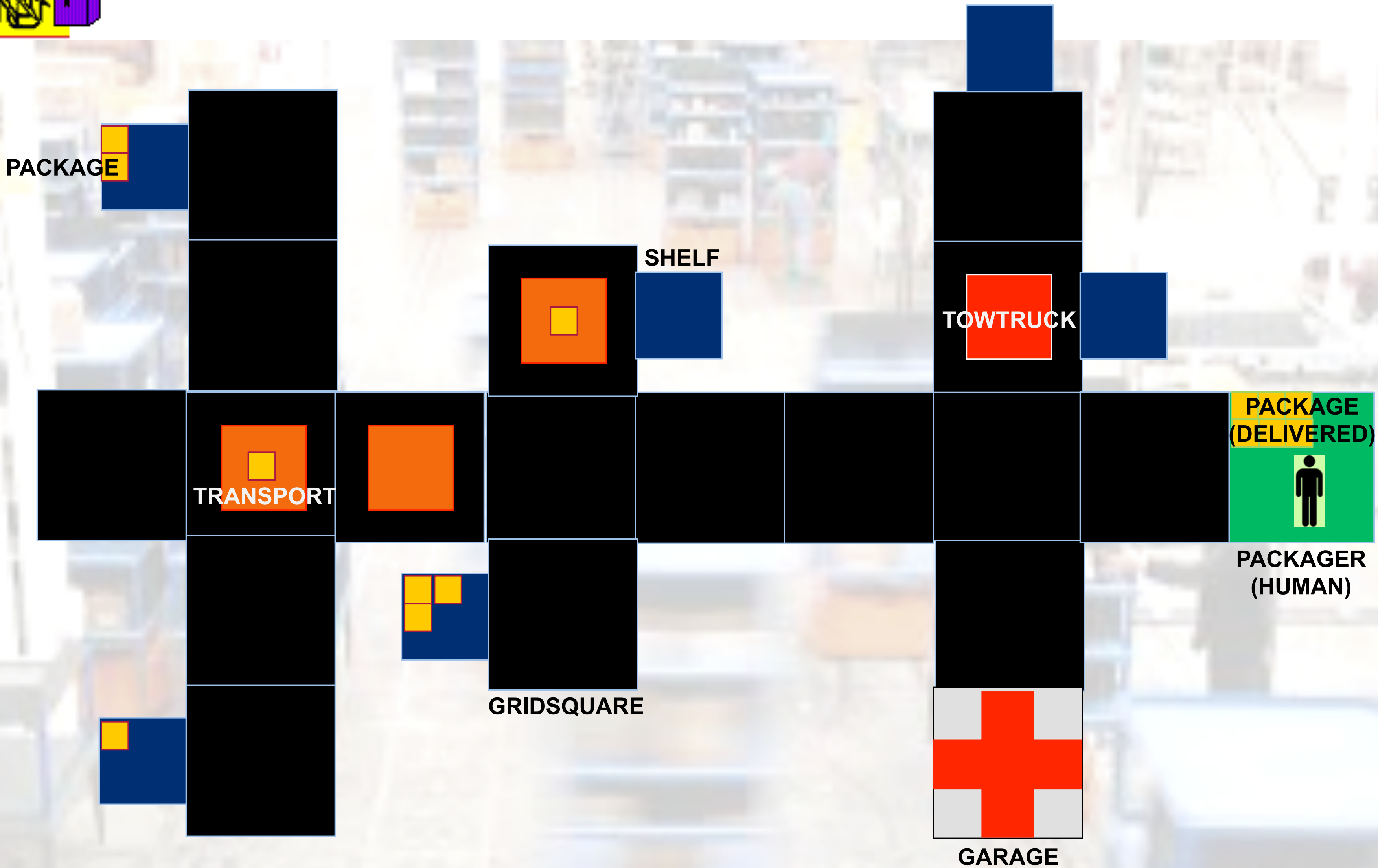
A Theory of Intra-Agent Replanning
Talamadupula, Smith, Cushing & Kambhampati

Replanning for Human-Robot Teaming

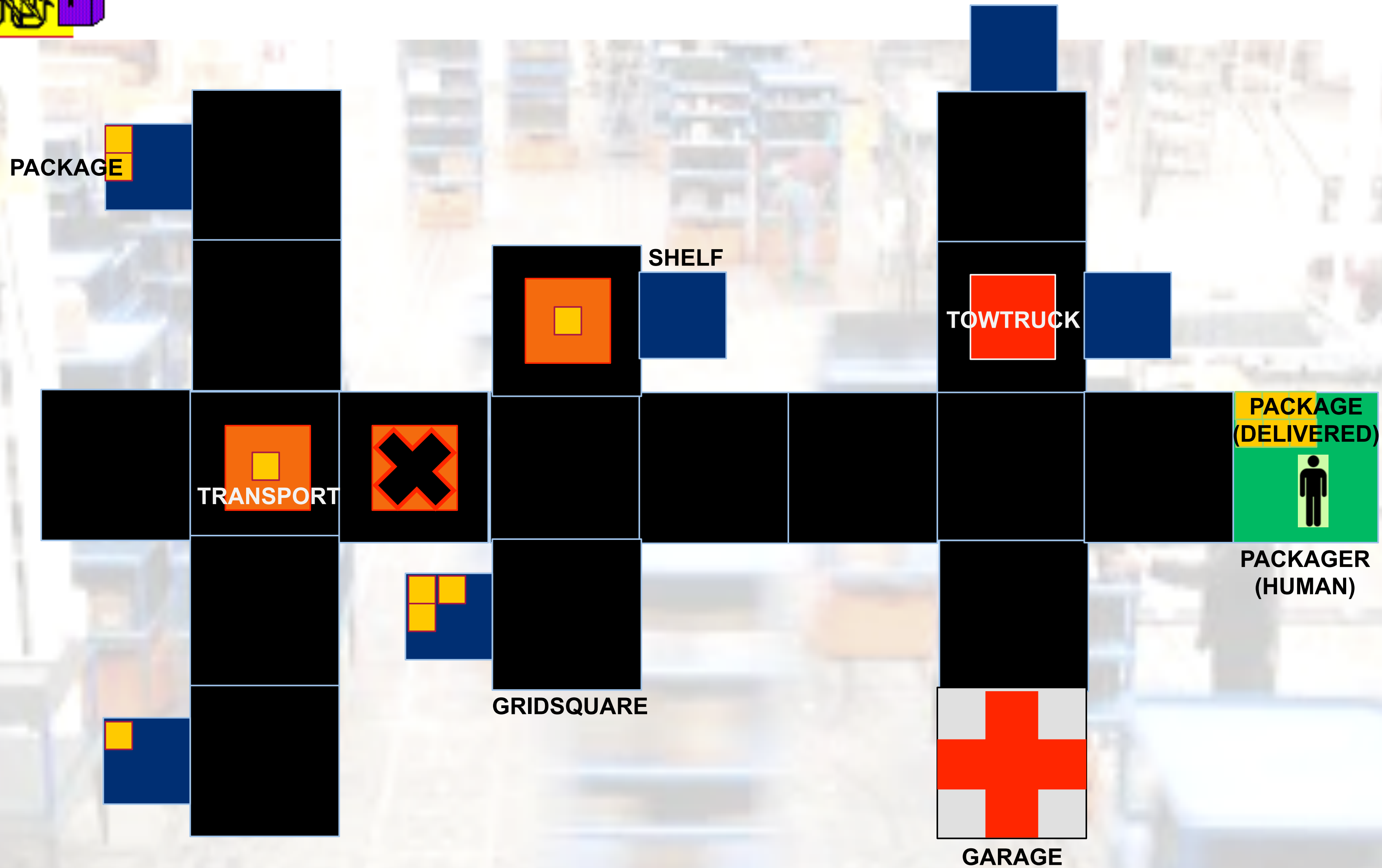
- Motivating Scenario: Automated Warehouses
  - Used by Amazon (Kiva Systems) for warehouse management
- Human: Packager
  - Only human on the entire floor; remotely located
  - Issues goals to the robotic agents
- Robot(s): Kiva Robots
  - Can transport items from shelves to the packager
- Goals: Order requests; come in dynamically
  - Goals keep changing as orders pile up
  - World changes as shelves are exhausted; break downs

[IROS09, AAAI10, TIST10, DMAP13, arXiv14]

Replanning Example: Warehouses

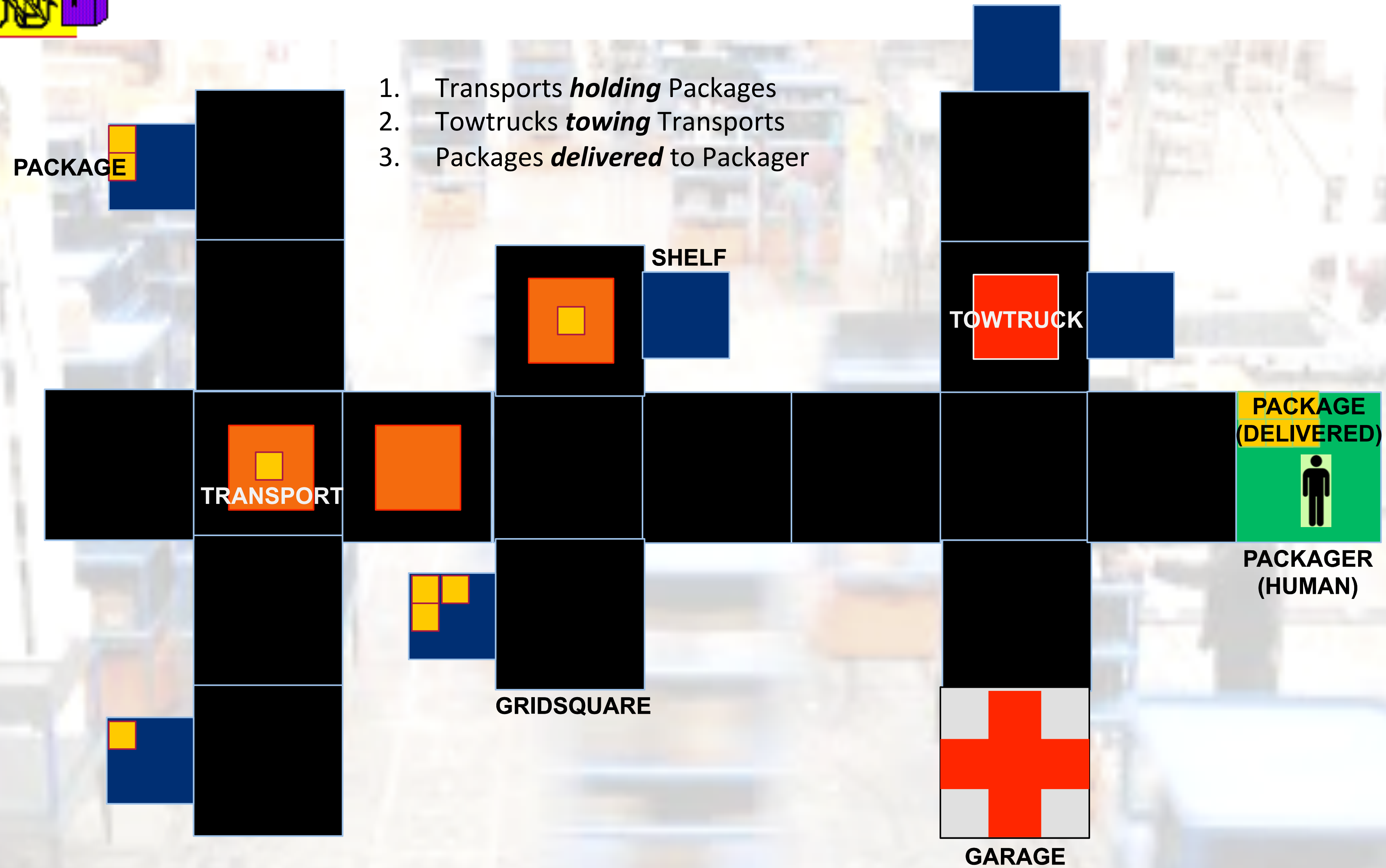


Warehouses: Perturbations



Warehouses: Commitments

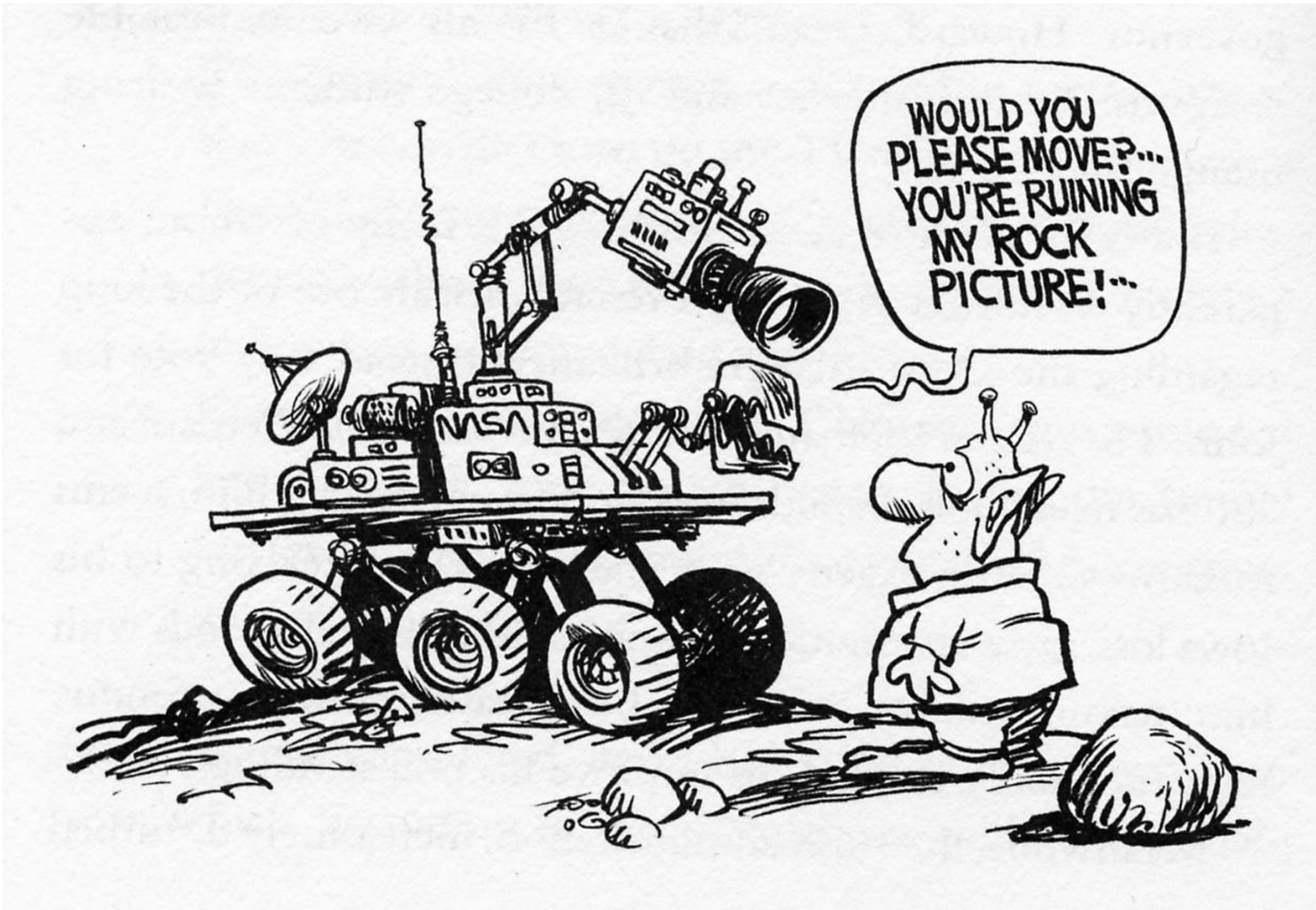
1. Transports ***holding*** Packages
2. Towtrucks ***towing*** Transports
3. Packages ***delivered*** to Packager



How to Replan

- *Abandon* Previous Plan
  - Discard old plan completely
- *Reuse* Previous Plan
  - Use information from π for generation of π'
  - Reuse parts of the original plan π
- *Commitments* from Previous Plan
  - What was previous plan achieving / promising?
    - Multi-agent: Inter-agent replanning problem produces intra-agent replanning problem
    - Project commitments made to other agents on to one's own planning process, as constraints

Commitments and Flexibility



Replanning Constraints

- **Unconstrained Replanning**
  - No constraints
- **Similarity Constrained Replanning**
  - Action Similarity
    - Minimize num of actions where π and π’ differ

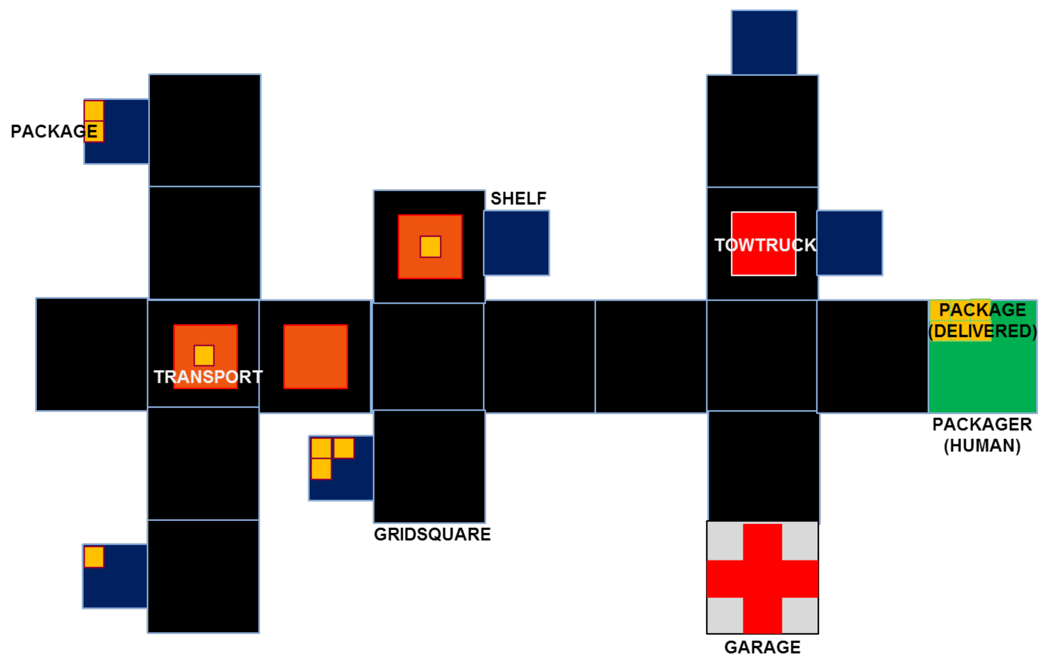
$$\mathbf{min}\ |\ \pi\ \Delta\ \pi'\ |$$

  - Causal Similarity
    - Minimize num of causal links where π and π’ differ

$$\mathbf{min}\ |\ CL(\pi)\ \Delta\ CL(\pi')\ |$$

Replanning Constraints

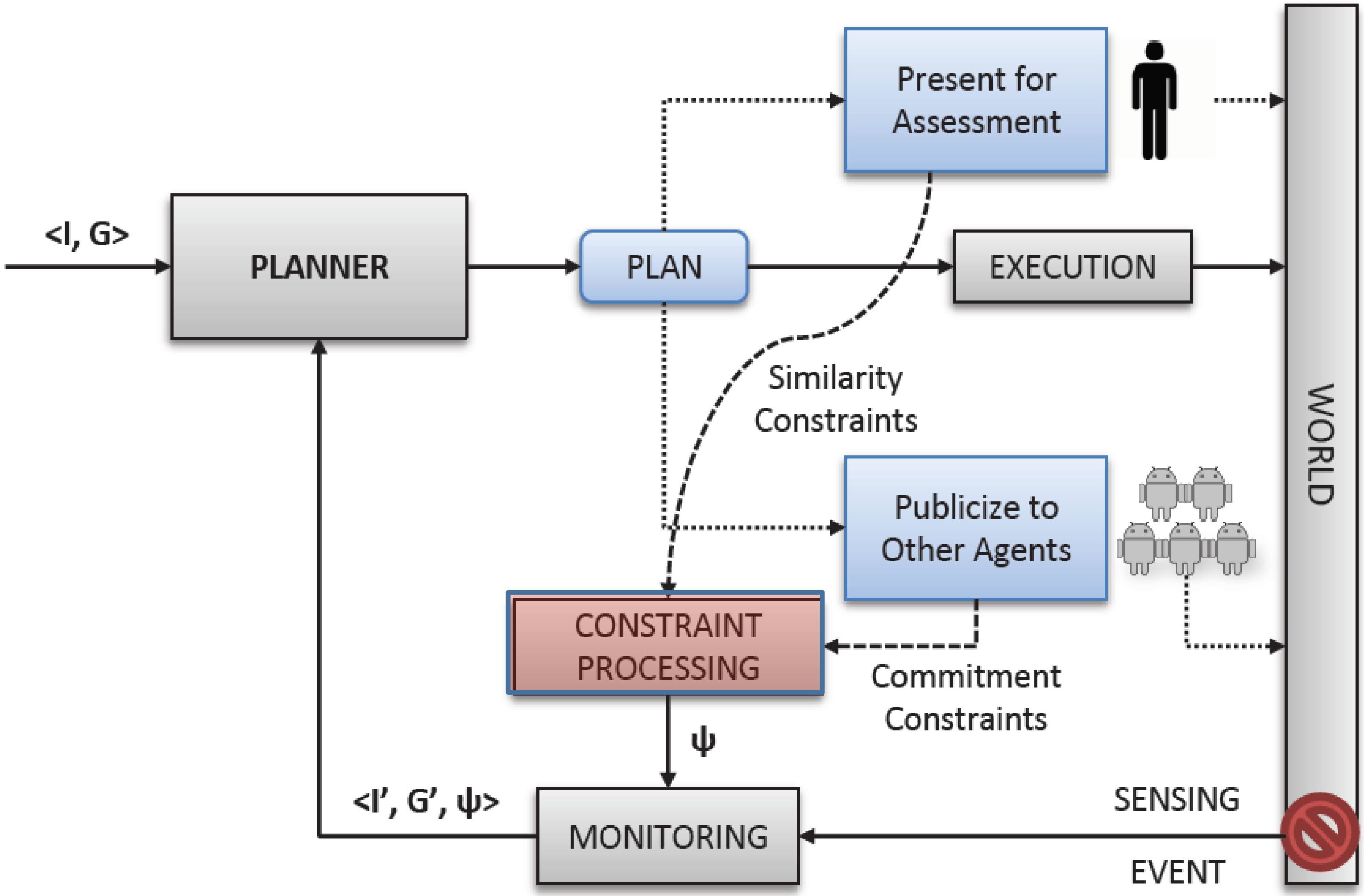
- **Commitment Constrained Replanning**
  - Dependencies between agents' plans
  - Project down into "commitments" between agents
    - Commitments depend on original plan π



Solution Techniques

- **Classical Planning**
  - Solve <**I’,G’**> using a classical planner
- **Specialized** Replanning Techniques
  - Iterative Plan Repair
    - Local Search
- **Compilation to Partial Satisfaction Planning**
  - *Commitments as constraints*
  - Model them as *soft goals*

A Generalized Model of Replanning



Replanning Constraints

M1 REPLANNING AS RESTART (From scratch)	› **No Constraints**
M2 REPLANNING AS REUSE (Similarity)	› Depends on the **similarity metric** between plans › **ACTION** SIMILARITY $\mathbf{min} \mid \pi \,\Delta\, \pi` \mid$ › **CAUSAL** SIMILARITY $\mathbf{min} \mid CL(\pi) \,\Delta\, CL(\pi`) \mid$
M3 REPLANNING TO KEEP COMMITMENTS	› **Dependencies** between π and other plans › Project down into **commitments** that π\` must fulfill › Exact nature of commitments depends on π › E.g.: **Multi-agent** commitments (between rovers)

Replanning: Solution Techniques

M1 REPLANNING AS RESTART (From scratch)	**CLASSICAL PLANNING**	› Solve new instance [I`,G`] for π` using classical planner
M2 REPLANNING AS REUSE (Similarity)	**ITERATIVE PLAN REPAIR** (Local Search)	› Start from π › Minimize differences while finding a candidate π` › Stop when [I`,G`] satisfied
M3 REPLANNING TO KEEP COMMITMENTS	**COMPILATION** (Partial Satisfaction Planning)	› Commitments are *constraints* on plan generation process › Commitments = Soft Goals G_s › Add G_s to G` → G`` › Run PSP planner with [I`,G``]

Commitments

- **All plans are made up of commitments**
  - Causal: Supporters provide conditions to consumers (partial order planning)
  - Agents: Enable / disable conditions for other agents
- **Commitments are also goals**
  - Most natural way of constraining change
  - Can model both "kinds" of replanning
    - **Commitment to Situation**: If a rover's view is blocked, must it uphold the commitment on observing?
    - **Commitment to Form**: If a rover's view is blocked, must it stick as close as possible to the previous plan?

Breaking Commitments

- Autonomous Robots (and planners)
  - Universal metric for defining penalty and reward values for commitments and goals
- Humans
  - Cannot reason with numbers alone
  - Need *explanations* or *excuses* on why commitments had to be broken
    - What explanations will a human accept?
    - Which excuses will make sense?
    - How can these be autonomously generated?

Rewards and Penalties

- A commitment is a soft goal
  - Reward r for fulfilling, penalty p for violating
    - Agents can give each estimates for r and p (summarization)
    - Else extract from a model of the other agent
- Action Similarity
  - For every action a in π, insert a goal
    - Any new plan π^* that contains fewer of π's actions than some other π° will have lesser net-benefit
      - Proof Sketch: In the absence of delete effects, π° can simply be a copy of π^* with one more action from π. If $r > p$, proved.
- Causal Similarity
  - For every causal link in π, insert a goal …

Rewards and Penalties

- How do we set values of r and p?
  - For agent-based commitments
    - From real-world applications
    - E.g.: NASA has estimates on how important certain observations or windows are (reward), and how costly missing them is (penalty)
    - New plan will take these into account (along with causal feasibility)
  - For similarity
    - Unit reward / penalty, in order to encourage more similarity with the previous plan π

Compiling Action Similarity to PSP

1. For every ground action **a** in π
   - Create a commitment constraint (goal) to have **a** in the new plan π`
     - Create predicate *a-executed* with the parameters* of **a**
     - Goal on *a-executed* (along with respective parameters*)
     - Assign unit reward and/or penalty to the goal

2. Add every goal thus generated to G` → G``

3. Give [I`,G``] to a PSP net-benefit planner
   - Return highest net-benefit plan as π`

* ground objects

[Talamadupula, Smith et al., DMAP 2013]

Compiling Causal Similarity to PSP

1. Obtain the relevant causal structure of π via regression

2. For every fluent f of every producer/consumer in that causal structure …
   - Create a commitment constraint (goal) to make π\` generate f
     - Create a predicate *f-link* with respective parameters, and a goal on it
     - Assign unit reward and/or penalty to the goal
   - Add every goal thus generated to G\` → G\`\`

3. Give [I\`,G\`\`] to a PSP net-benefit planner
   - Return highest net-benefit plan as π\`

[Talamadupula, Smith et al., DMAP 2013]

Compiling Similarity to PSP

ACTION SIMILARITY

prec(a)

a

eff(a)

prec(a)

a_{as}

eff(a)

a-executed

CAUSAL SIMILARITY

prec(a)

a

eff(a)

for all f in prec(a) s.t. f is the reason a is a consumer, f-link

prec(a)

a_{cs}

eff(a)

for all f in eff(a) s.t. f is the reason a is a producer, f-link

Replanning Constraints

M1 REPLANNING AS RESTART (From scratch)	› **No Constraints**
M2 REPLANNING AS REUSE (Similarity)	› Depends on the **similarity metric** between plans › **ACTION** SIMILARITY $\mathbf{min} \mid \pi \,\Delta\, \pi` \mid$ › **CAUSAL** SIMILARITY $\mathbf{min} \mid CL(\pi) \,\Delta\, CL(\pi`) \mid$
M3 REPLANNING TO KEEP COMMITMENTS	› **Dependencies** between π and other plans › Project down into **commitments** that π` must fulfill › Exact nature of commitments depends on π › E.g.: **Multi-agent** commitments (between rovers)

Motivation

Grandpa Hates Robots
Köckemann, Karlsson & Pecora

- Task planning in inhabited environments
  (aka Human-aware Task Planning)
- Humans impose rules on acceptability of plans
  → "Grandpa hates robots"
  → "Don't vaccuum while I'm reading"
  → "Don't enter the bathroom when it's occupied"

Samuel Goldwyn Films, Robot & Frank (2012)

Interaction Constraints for Planning in Inhabited Environments

- Extending constraint-based planning with human-awareness
- Domain models contain variety of constraint types
  $\rightarrow$ Temporal constraints, resources, goals, costs, Prolog, ...
- Contribution: Interaction constraints (ICs)
  $\rightarrow$ Modeling interactions between humans and robots
- Examples:
  $\rightarrow$ "Don't vacuum while I'm reading"
  $\rightarrow$ "Don't enter the bathroom when it's occupied"
- Can handle partially specified human activities

OVERVIEW

1. INTRODUCTION
2. INTERPRETATION
3. DECISION SUPPORT
   a. Explicit Constraints
   b. Implicit Constraints (Preferences)
   c. Incomplete Dynamics
4. COMMUNICATION
   a. Excuses & Explanations
   b. Asking for Help
5. CASE STUDY
6. SUMMARY

A little recap..

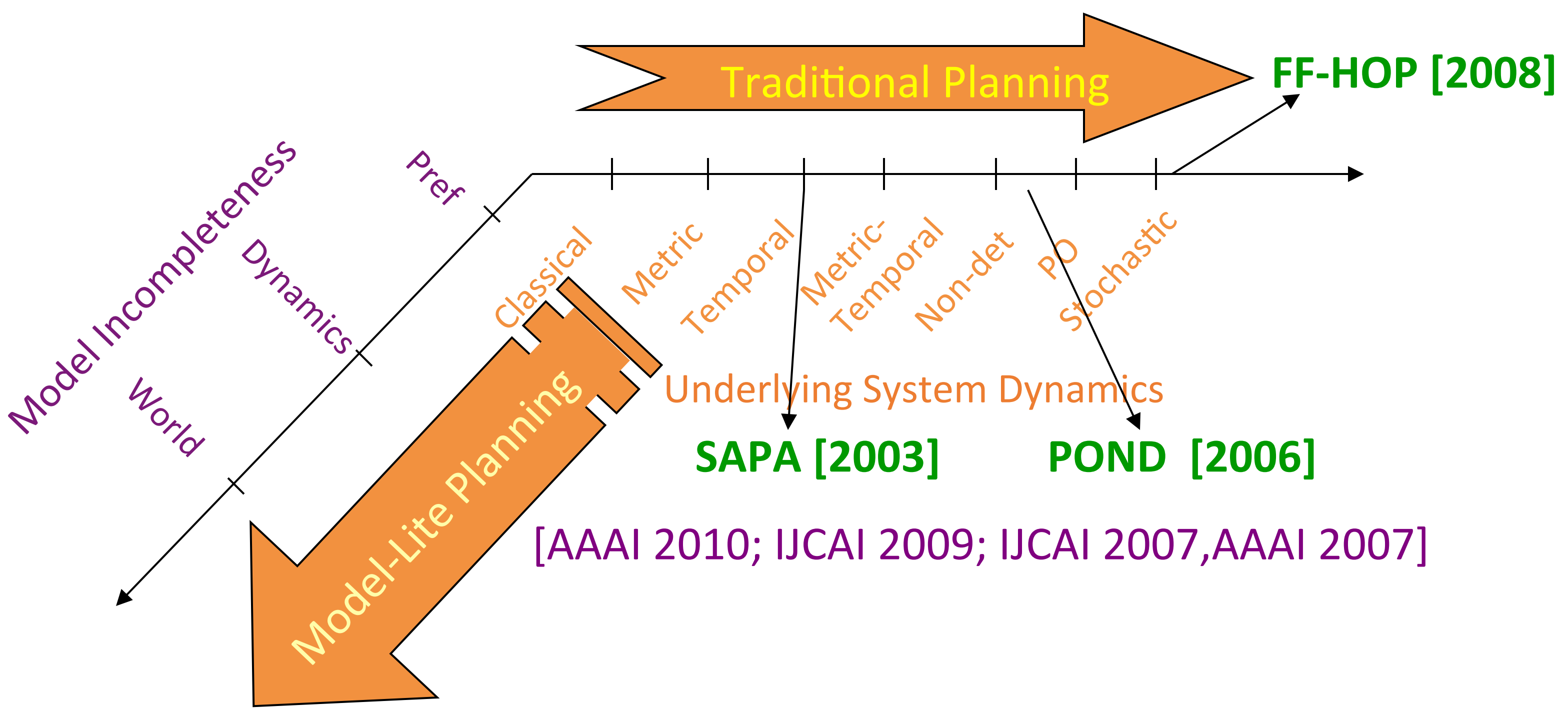
- Until now we have been adapting current planning techniques for

Assumption: Complete Models

- →~~Complete~~ Action Descriptions (fallible domain writers)
- →~~Fully Specified~~ Preferences (uncertain users)
- →All ~~objects~~ in the world known up front (open worlds)
- →~~One-shot~~ planning (continual revision)

Planning is no longer a pure inference problem ☹

☹ But humans in the loop can ruin a really a perfect day ☹



Effective ways to handle the more expressive planning problems by exploiting the deterministic planning technology

TRAINS—planner with limited capabilities

Interestingly perhaps, traditional planning technology does not play a major role in the system, and in fact it is difficult to see how such components might fit into a mixed-initiative system. We describe some of these

Although it's doing planning, however, a mixed-initiative planning system isn't doing what we might recognize as "traditional" planning, that is, constructing a sequence of operators from a fully-specified initial situation to a stated goal. In fact, in an informal analysis of one hour of human-human problem-solving dialogues (part of a larger eight hour study (Heeman & Allen 1995)), we found that a relatively small percentage of the utterances, 23%, dealt with explicitly adding or refining actions in the plan. Figure 4 summarizes this analysis. Note the importance of being

Evaluation/comparison of options	25%
Suggesting courses of action	23%
Establishing world state	13%
Clarifying/confirming communication	13%
Discussing problem solving strategy	10%
Summarizing courses of action	10%
Identifying problems/alternatives	7%

ager had stated their goals. However, not only is it unlikely that we will ever be able to build such a reasoner for a realistic domain, in the next section we claim that such a system is not necessarily appropriate for mixed-initiative planning. We therefore deliberately weakened the TRAINS-95 domain reasoner in order to force the manager to interact in order to overcome its shortcomings. The route planner can therefore only plan route segments less than four hops long, and for those it chooses a random path. The knowledge base maintains an accurate view of the map, and allows various "natural" events such as bad weather or track maintenance to arise during the interaction. These also force interaction in order to revise plans to take account of them.

Learning is not the (sole) answer..

- A tempting way to handle incompleteness is to say that we should wait until the full model is obtained
  - Either through learning
  - Or by the generosity of the domain writer..
- Problem: Waiting for complete model is often times not a feasible alternative
  - The model may never become complete…
  - We need to figure out a way of maintaining incomplete models, and planning with them (pending learning..)

Challenges of Handling Incompleteness

1. Circumscribing the incompleteness

2. Developing the appropriate solution concepts

3. Developing planners capable of synthesizing them

4. Life-long Planning/Learning to reduce incompleteness
   - Commitment-sensitive Replanning

Preferences in Planning – Traditional View

- Classical Model: "Closed world" assumption about user preferences.
  - All preferences assumed to be fully specified/available

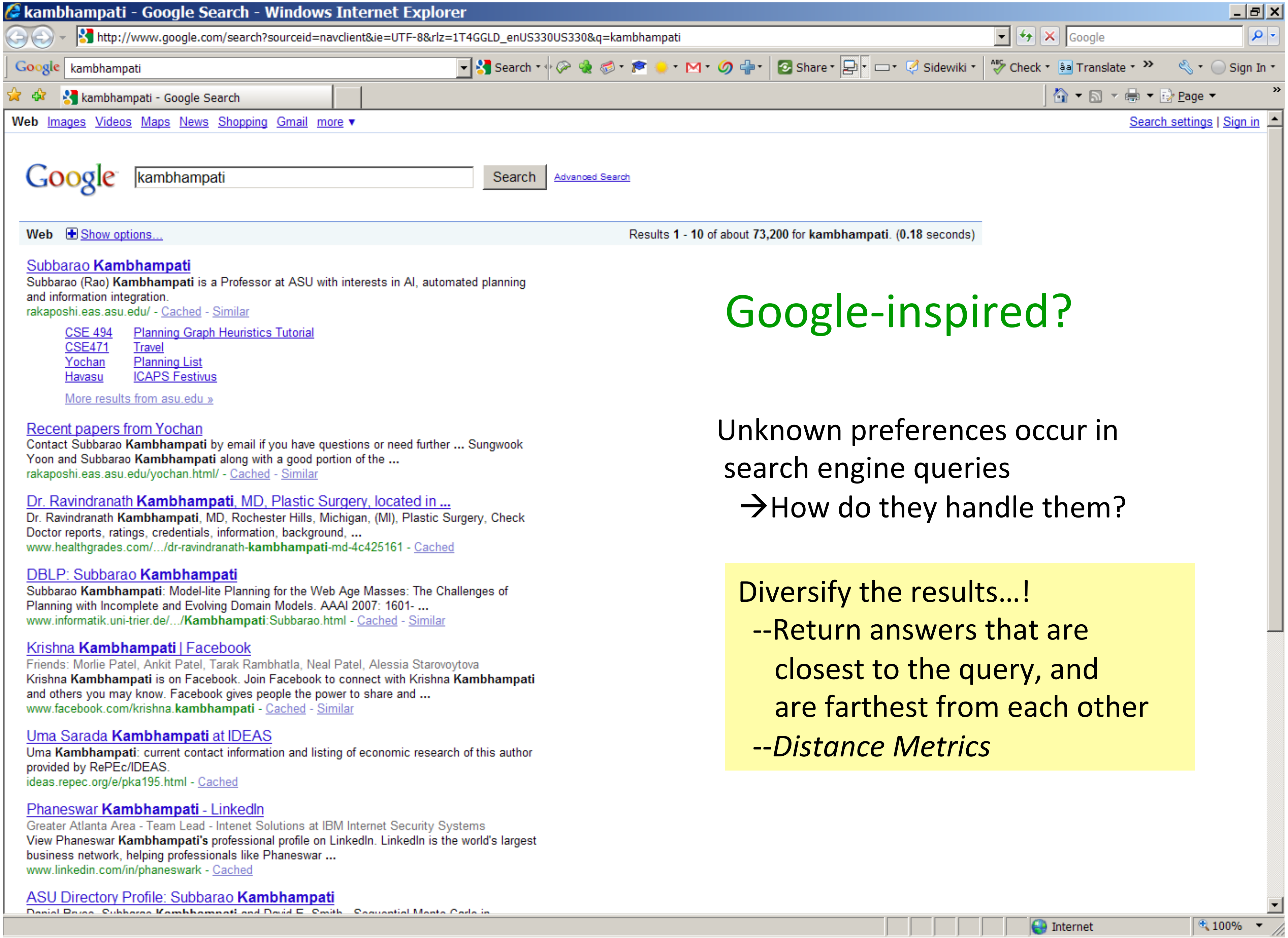
Full Knowledge of Preferences

Two possibilities

- If no preferences specified —then user is assumed to be *indifferent*. Any single feasible plan considered acceptable.
- If preferences/objectives are specified, find a plan that is optimal w.r.t. specified objectives.

Either way, solution is a *single* plan

Human in the Loop: Unknown & Partially Known Preferences



Google-inspired?

Unknown preferences occur in search engine queries

→How do they handle them?

Diversify the results…!

--Return answers that are closest to the query, and are farthest from each other

--*Distance Metrics*

Handling Unknown & Partially Known Preferences

- **Unknown preferences**
  - For all we know, user may care about every thing -- the flight carrier, the arrival and departure times, the type of flight, the airport, time of travel and cost of travel…
  - Best choice is to return a *diverse* set of plans [IJCAI 2007]
    - Distance measures between plans

Generating Diverse Plans

- *d*DISTANT*k*SET
  - Given a distance measure δ(.,.), and a parameter k, find k plans for solving the problem that have guaranteed minimum pair-wise distance *d* among them in terms of δ(.,.)

- Formalized notions of bases for plan distance measures
- Proposed adaptation to existing representative, state-of-the-art, planning algorithms to search for diverse plans
  - Showed that using action-based distance results in plans that are likely to be also diverse with respect to behavior and causal structure
  - LPG can scale-up well to large problems with the proposed changes

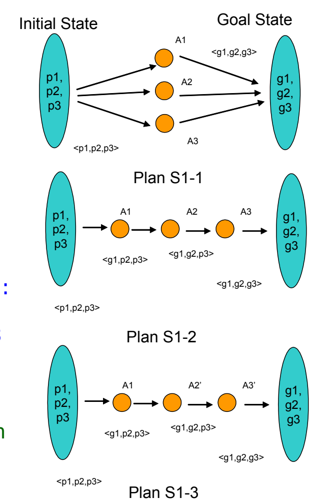
[IJCAI 2007]

Distance Measures

- In what terms should we measure distances between two plans?
  - The actions that are used in the plan?
  - The behaviors exhibited by the plans?
  - The roles played by the actions in the plan?
- Choice may depend on
  - The ultimate use of the plans
    - E.g. Should a plan P and a non-minimal variant of P be considered similar or different?
  - What is the source of plans and how much is accessible?
    - E.g. do we have access to domain theory or just action names?

Compute by Set-difference

- •Action-based comparison: S1-1, S1-2 are similar, both dissimilar to S1-3; with another basis for computation, all can be seen as different
- •State-based comparison: S1-1 different from S1-2 and S1-3; S1-2 and S1-3 are similar
- •Causal-link comparison: S1-1 and S1-2 are similar, both diverse from S1-3



Generating Diverse Plans with Local Search

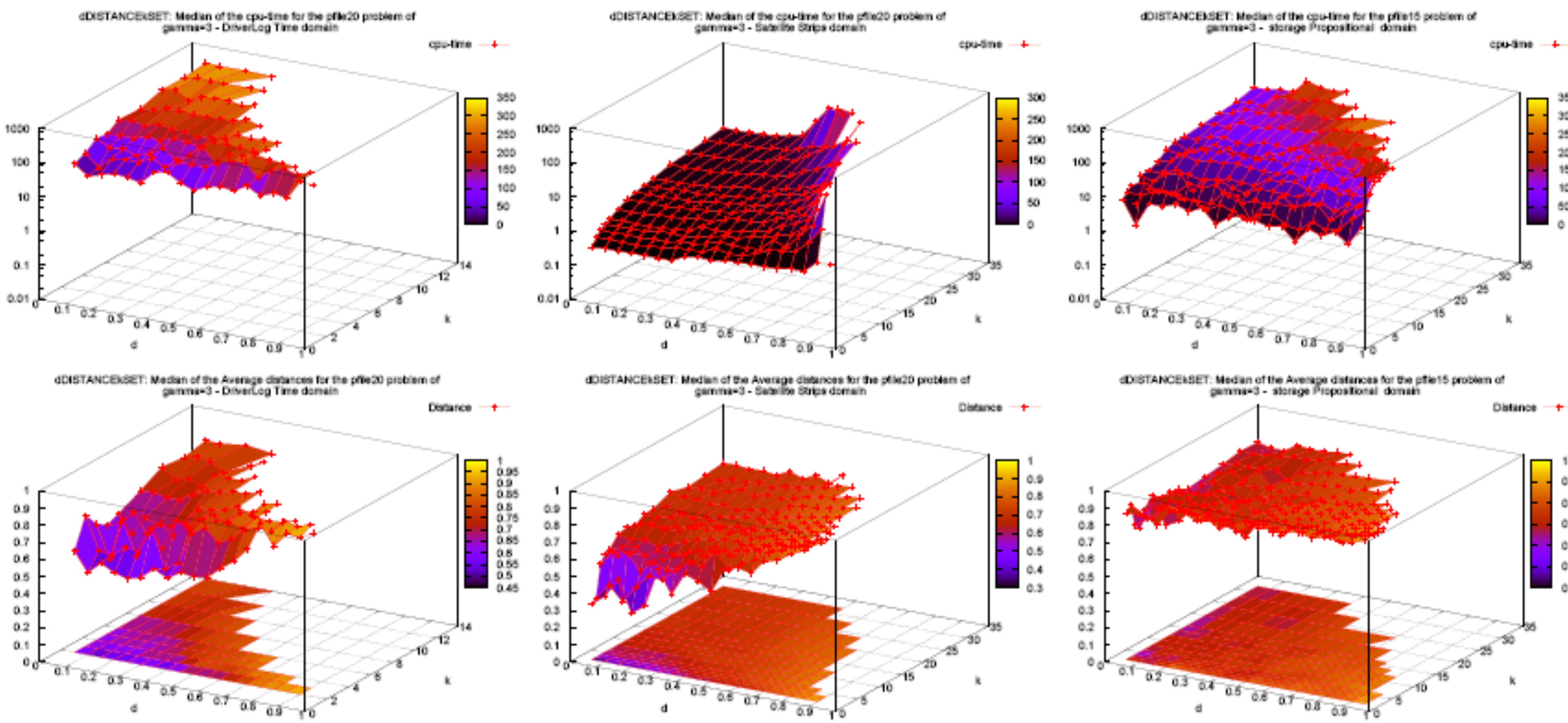


Figure 2: Performance of LPG-d (CPU-time and plan distance) for there problems in DriverLog-Time, Satellite-Strips and Storage-Propositional.

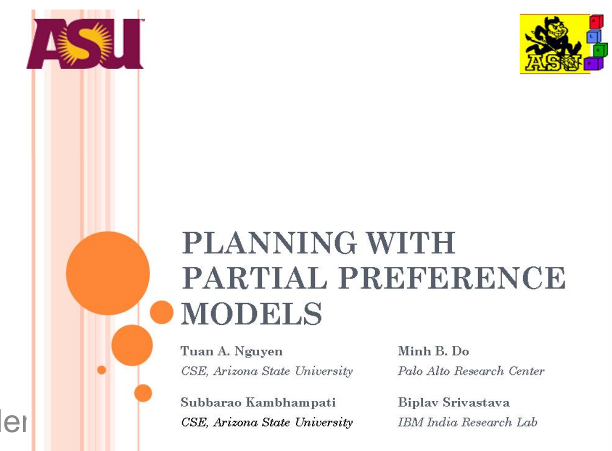
LPG-d solves 109 comb.
Avg. time = 162.8 sec
Avg. distance = 0.68
Includes d<0.4,k=10; d=0.95,k=2

LPG-d solves 211 comb.
Avg. time = 12.1 sec
Avg. distance = 0.69

LPG-d solves 225 comb.
Avg. time = 64.1 sec
Avg. distance = 0.88

Unknown & Partially Known Preferences

- **Partially known**
  - We may know that user cares only about makespan and cost. But we don't know how she combines them..
- Returning a diverse set of plans may not be enough
  - *They may not differ on the attributes of relevance..*
- Focus on spanning the pareto set..



Modeling Partially Known Objectives

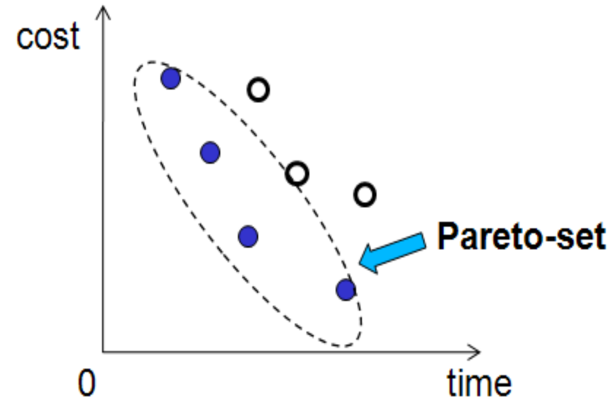
- The user is interested in minimizing two objectives (say makespan and execution cost of plan *p*: *time*(*p*), *cost*(*p*).)
- The quality of plan *p* is given by *cost function*:

  $$f(p,w) = w \times time(p) + (1-w) \times cost(p) \quad (w \in [0,1])$$

  - $w \in [0,1]$ represents the trade-off between two competing objectives.

measure developed in OR community (Carlyle, 2003).

Handling Partially Known Preferences

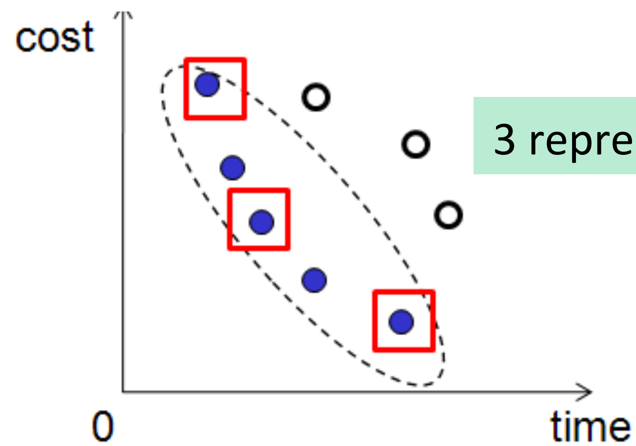


- **View it as a Multi-objective optimization**
  - Return the Pareto optimal set of plans (and let the user select from among them)
- Two problems
  - [Computational] Computing the full pareto set can be too costly
  - [Comprehensional] Lay users may suffer information overload when presented with a large set of plans to choose from
- Solution: Return *k* representative plans from the Pareto Set
  - Challenge 1: How to define "representative" robustly?
  - Challenge 2: How to generate representative set of plans efficiently?

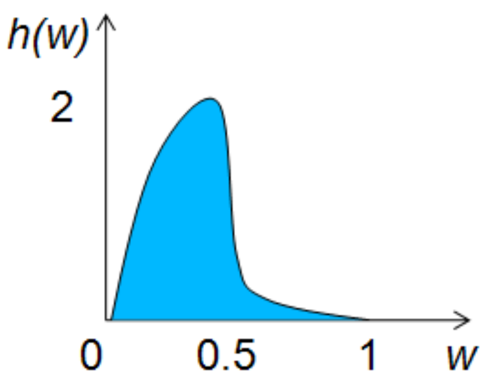
Measuring Representativeness: ICP

$$f(p,w) = w \times time(p) + (1-w) \times cost(p) \quad (w \in [0,1])$$

$$ICP(\mathcal{P}) = \sum_{i=1}^{k} \int_{w_{i-1}}^{w_i} h(w)(w \times t_{p_i} + (1-w) \times c_{p_i})dw$$



3 representative plans



(Fast plan seems better)

Handling Partial Preferences using ICP

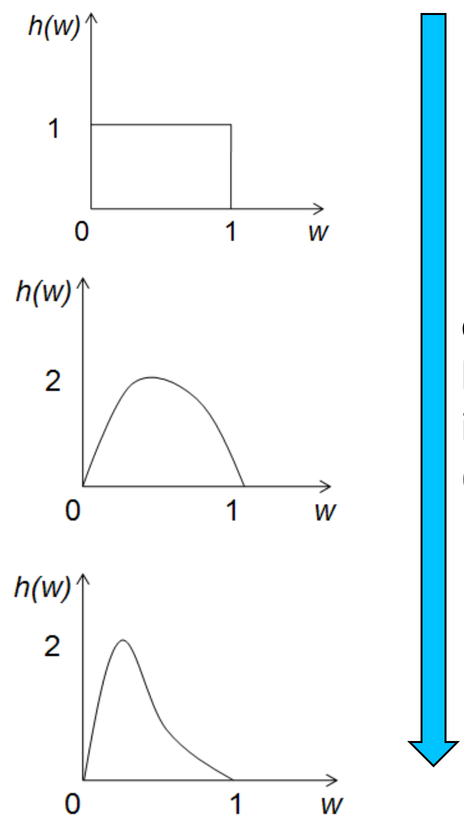
Problem Statement:

- Given
  - the objectives O_i,
  - the vector w for convex combination of O_i
  - the distribution $h(w)$ of w,
- Return a set of k plans with the minimum ICP value.

- Solution Approaches:
  - Sampling: Sample k values of w, and approximate the optimal plan for each value.
  - ICP-Sequential: Drive the search to find plans that will improve ICP
  - Hybrid: Start with Sampling, and then improve the seed set with ICP-Sequential
  - [Baseline]: Find k diverse plans using the distance measures from [IJCAI 2007] paper; LPG-Speed.

Learning Planning Preferences

- We can learn to improve the preference model by revising the *h(w)* after every few iterations (through user interaction)

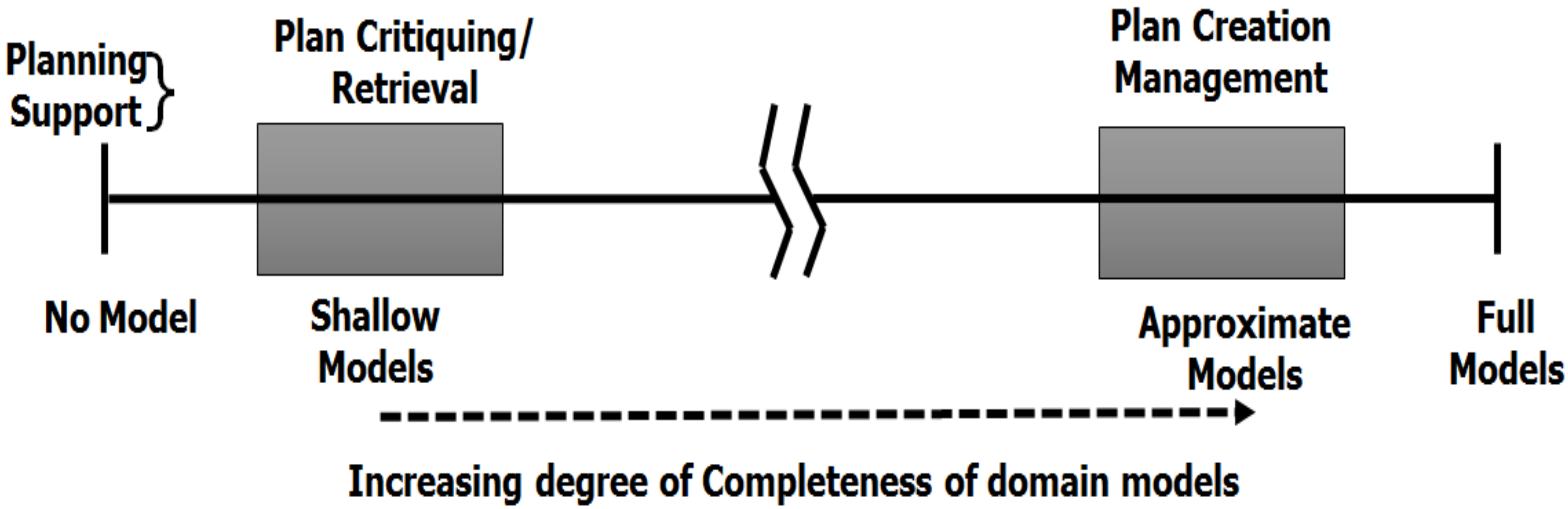


Revising distribution h(w) over iterations (Bayesian learning..)

OVERVIEW

1. INTRODUCTION
2. INTERPRETATION
3. DECISION SUPPORT
   a. Explicit Constraints
   b. Implicit Constraints (Preferences)
   c. Incomplete Dynamics
4. COMMUNICATION
   a. Excuses & Explanations
   b. Asking for Help
5. CASE STUDY: HUMAN-ROBOT TEAMING
6. SUMMARY

Models v. Planning Capabilities



I/O types
Task dependency
(e.g. workflows management, web service composition)

Missing some preconditions/ effects of actions
(e.g. Garland & Lesh, 2002)

There are known knowns; there are things we know that we know. There are known unknowns; that is to say, there are things that we now know we don't know. But there are also unknown unknowns; there are things we do not know we don't know.

Approaches for Planning with Incomplete Models

Incompleteness annotations are available

- An alternative way to make-up for model incompleteness is to expect annotations circumscribing the extent of incompleteness
- In this case, we can explicitly reason with the correctness of candidate plans over all possible models
  - Nguyen et. Al NIPS 2013; ICAPS 2014; Weber & Bryce, ICAPS 2011

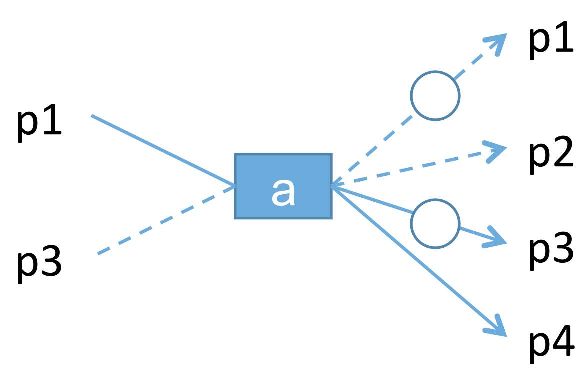
Library of cases is available

- ML-CBP exploits cases directly during planning (by transferring case fragments into a skeletal plan generated w.r.t. M')
  - AAAI 2013
- An alternative approach would be to use the cases C to *refine* the model M' into a more accurate model M'' (where M'' is a better approximation of M*). Come see our IJCAI 2013 paper.
  - Zhuo et. Al. IJCAI 2013
  - M'' contains both primitive and macro- operators

Deterministic Partial Domain Models

- We consider planning with deterministic, but incompletely specified domain model
- Each action **a** is associated with *possible* precond and effects (in addition to the normal precond/eff):
  - **PreP(a)** *[p]:* set of propositions that **a** *might* depend on during execution
  - **AddP(a)** *[p]:* : set of propositions that **a** *might* add after execution
  - **DelP(a)** *[p]:* : set of propositions that **a** *might* delete after execution

Example: An action **a** that is known to depend on **p1**, add p4 and delete **p3**. In addition, it might have **p3** as its precondition, might add **p2** and might delete **p1** after execution.



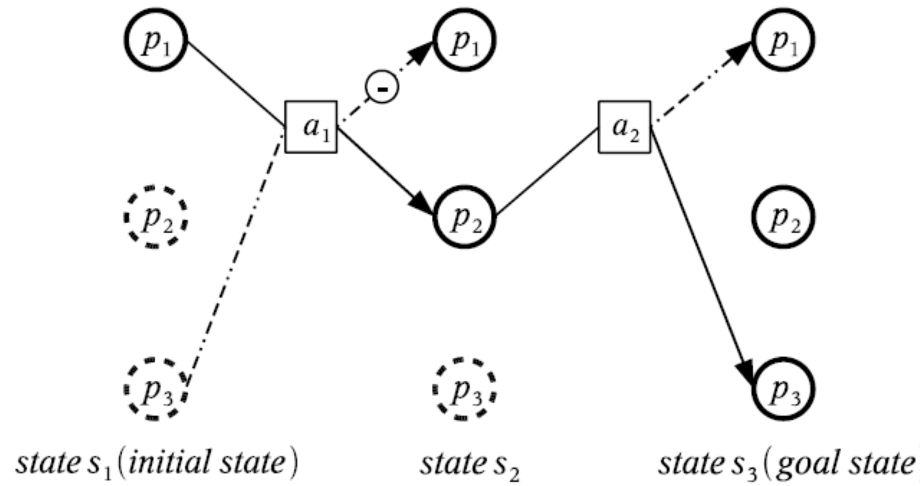
Solution Concept: Robust Plans

- Solution concept:
  - Robust plan
  - Plan is highly robust if executable in large number of most-likely candidate models
- Robustness measure
  - Set of candidate domain models **S** (consistent with the given deterministic partial domain model **D**)
  - A complete but unknown domain model **D***
    - Can be any model in **S**

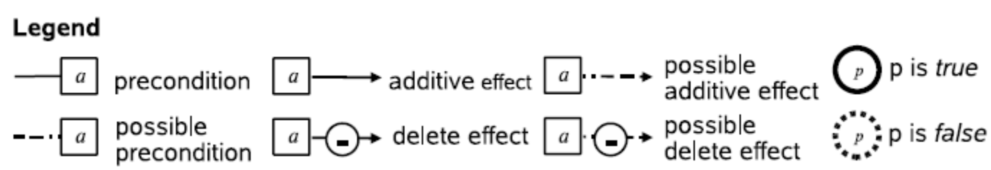
$$R(\pi) = \frac{|\Pi|}{2^K}$$

$|\Pi|$ Number of candidate models with which the plan succeeds

$$K = \sum_a \mathrm{PreP}(a) + \mathrm{AddP}(a) + \mathrm{DelP}(a)$$



Candidate models of plan	1	2	3	4	5	6	7	8
a_1 relies on p_3	yes	yes	yes	yes	no	no	no	no
a_1 deletes p_1	yes	yes	no	no	yes	yes	no	no
a_2 adds p_2	yes	no	yes	no	yes	no	yes	no
Plan status	fail	fail	fail	fail	succeed	fail	succeed	succeed



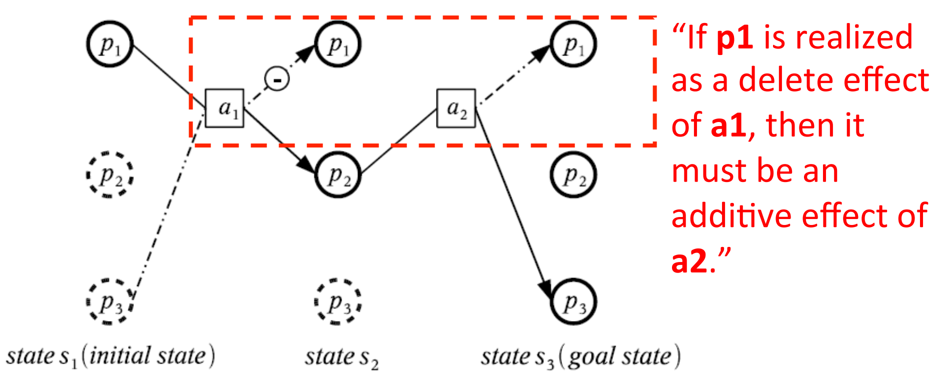
Robustness value: 3/8

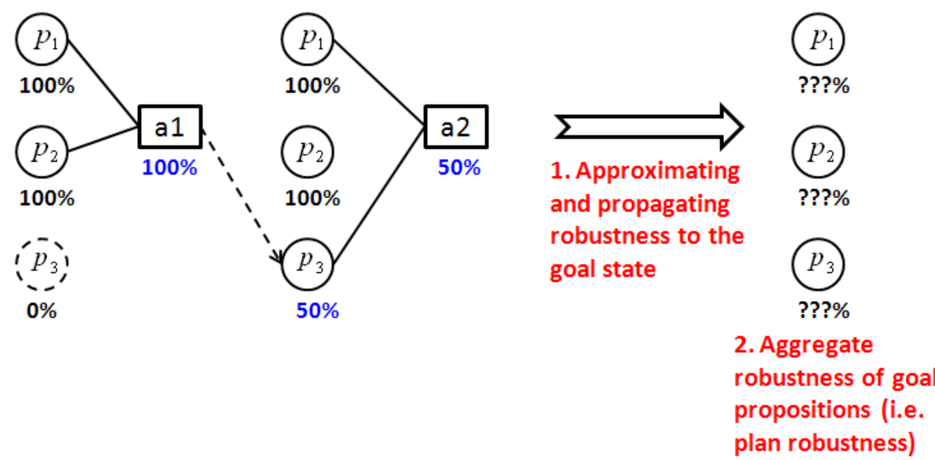
Easily generalized to consider model likelihood

Assessing Plan Robustness

- Number of candidate models: exponentially large. Computing robustness of a given plan is hard!!!
  - Exact and approximate assessment.
- **Exact methods:**
  - (Weighted) Model-counting approach:
    - Construct logical formulas representing *causal-proof* (Mali & Kambhampati 1999) for plan correctness
    - Invoke an exact model counting approach

- **Approximate methods:**
  – Invoke *approximate* model counting approach
  – Approximate and propagate action robustness
    - Can be used in generating robust plans





- **Plan correctness constraints Σ**
  - **Establishment constraints**

$$\bigvee_{C_p^i \le k \le i-1, p \in \widetilde{Add}(a_k)} p_{a_k}^{add} \qquad p_{a_i}^{pre} \Rightarrow \bigvee_{C_p^i \le k \le i-1, p \in \widetilde{Add}(a_k)} p_{a_k}^{add}$$

  - **Protecting constraints**

$$p_{a_m}^{del} \Rightarrow \bigvee_{C_p^i \le k \le i-1, p \in \widetilde{Add}(a_k)} p_{a_k}^{add}$$

$$p_{a_i}^{pre} \Rightarrow (p_{a_m}^{del} \Rightarrow \bigvee_{C_p^i \le k \le i-1, p \in \widetilde{Add}(a_k)} p_{a_k}^{add})$$

Plan robustness
≡
Weighted model counting WMC(Σ)

Monotone clauses, but exact WMC(Σ) is provably costly!

Generating Robust Plans

- Compilation approach: Compile into a *(Probabilistic) Conformant Planning* problem
  - One "unobservable" variable per each possible effect/precondition
    - Significant initial state uncertainty
  - Can adapt a probabilistic conformant planner such as POND [JAIR, 2006; AIJ 2008]
- Direct approach: Bias a planner's search towards more robust plans
  - Heuristically assess the robustness of partial plans
    - Need to use the (approximate) robustness assessment procedures

D. Bryce et al. / Artificial Intelligence 172 (2008) 685–715

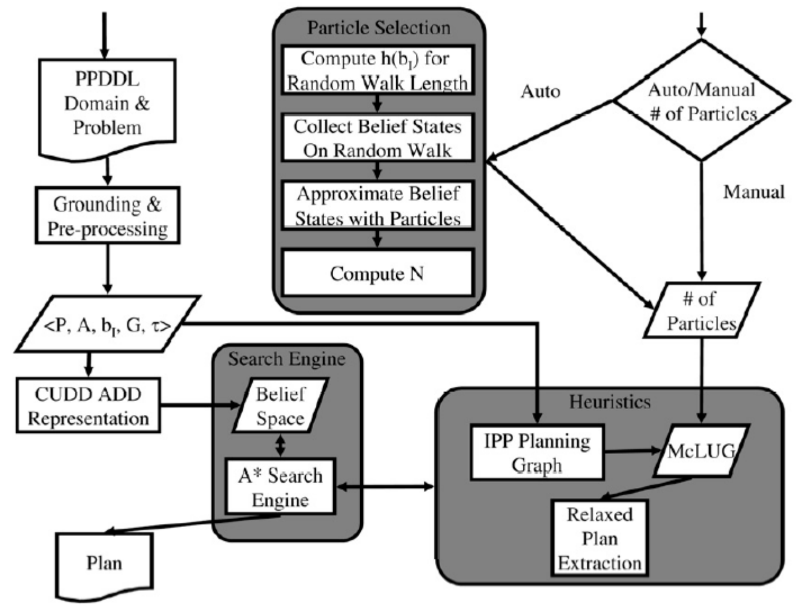
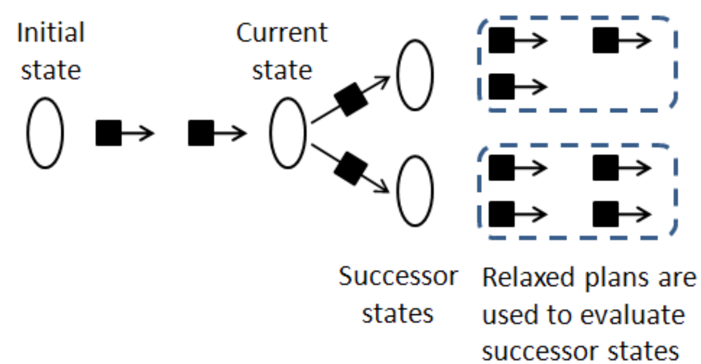


Fig. 6. *POND* architecture.



[Bryce et. Al., ICAPS 2012; Nguyen et al; NIPS 2013; Nguyen & Kambhampati, ICAPS 2014]

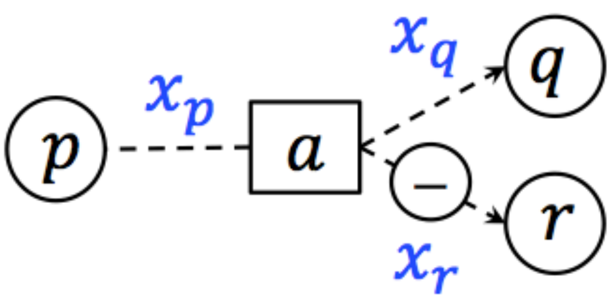
Synthesizing Robust Plans: A Compilation

Incomplete model
Complete world state

→

Complete model
Belief state

(Conformant Probabilistic Planning)



x_p (0.5) x_q (0.7) x_r (0.2)

Resulting action $\boldsymbol{a}'$ with eight conditional effects.

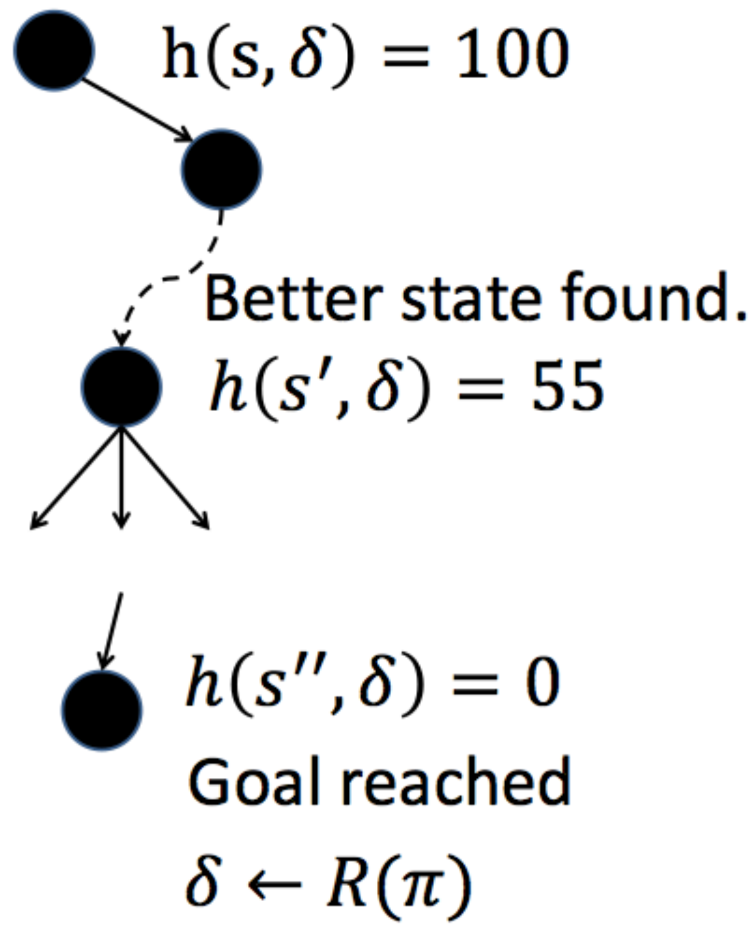
Cond: $x_p \wedge p \wedge x_q \wedge x_r$ **Eff:** $q \wedge \neg r$

Synthesizing Robust Plans: A Heuristic Search

❖ **Anytime approach**

1. Initialize: $\delta = 0$
2. **Repeat**
   - ❖Find plan π s.t. $R(\pi) > \delta$
   - ❖If plan found: $\delta = R(\pi)$

   Until time bound reaches
3. Return π and $R(\pi)$ if plan found



$h(s, \delta)$: how far it is approximately from s to a goal state so that the resulting plan has approximate robustness > δ.

- **Approximate plan robustness**
  - **Lower bound**

$$l(\Sigma) = \prod_{c \in \Sigma} \Pr(c) \;\leq\; WMC(\Sigma)$$

$$\overset{I}{\bullet} \xrightarrow[\pi_k]{\Sigma_{\pi_k}} \bullet \xrightarrow[\text{Relaxed plan } \tilde{\pi}]{\Sigma_{\tilde{\pi}}} \bullet \qquad l\left(\Sigma_{\pi_k} \wedge \Sigma_{\tilde{\pi}}\right) > \delta$$

  - **Upper bound**: divide Σ into independent sets Σ^{i}

$$u(\Sigma) = \prod_{\Sigma^i} \min_{c \in \Sigma^i} \Pr(c) \geq WMC(\Sigma)$$

$$\overset{I}{\bullet} \xrightarrow[\pi_k]{\Sigma_{\pi_k}} \bullet$$

If $u\left(\Sigma_{\pi_k}\right) > \delta$

then compute $WMC(\pi_k)$

Approaches for Planning with Incomplete Models

Incompleteness annotations are available

- An alternative way to make-up for model incompleteness is to expect annotations circumscribing the extent of incompleteness
- In this case, we can explicitly reason with the correctness of candidate plans over all possible models
  - Nguyen et. Al NIPS 2013; ICAPS 2014; Weber & Bryce, ICAPS 2011

Library of cases is available

- ML-CBP exploits cases directly during planning (by transferring case fragments into a skeletal plan generated w.r.t. M')
  - AAAI 2013
- An alternative approach would be to use the cases C to *refine* the model M' into a more accurate model M'' (where M'' is a better approximation of M*). Come see our IJCAI 2013 paper.
  - Zhuo et. Al. IJCAI 2013
  - M'' contains both primitive and macro- operators

OVERVIEW

1. INTRODUCTION
2. INTERPRETATION
3. DECISION SUPPORT
    a. Explicit Constraints
    b. Implicit Constraints (Preferences)
    c. Incomplete Dynamics
4. COMMUNICATION
    a. Excuses & Explanations
    b. Asking for Help
5. CASE STUDY: HUMAN-ROBOT TEAMING
6. SUMMARY

OVERVIEW

1. INTRODUCTION
2. INTERPRETATION
3. DECISION SUPPORT
   a. Explicit Constraints
   b. Implicit Constraints (Preferences)
   c. Incomplete Dynamics
4. COMMUNICATION
   a. Excuses & Explanations
   b. Asking for Help
5. CASE STUDY
6. SUMMARY

Failures in Planner-Based Systems

When acting in a uncertain, dynamic environment, things can go wrong:

- Execution failures
  - Error diagnosis
  - Continual Planning
- Planning failures
  - Domain is incorrectly modelled
  - Incomplete world knowledge
  - Missing resources
  - Maybe the task is just unsolvable

Coming up With Good Excuses
Göbelbecker, Keller, Eyerich, Brenner & Nebel (2010)

Explaining Planning Failures

An excuse is a *counterfactual statement* about the planning problem:

> "If the door were unlocked, then I could find a plan to bring you the coffee and the newspaper."

What is an Excuse?

Definition (Excuse)

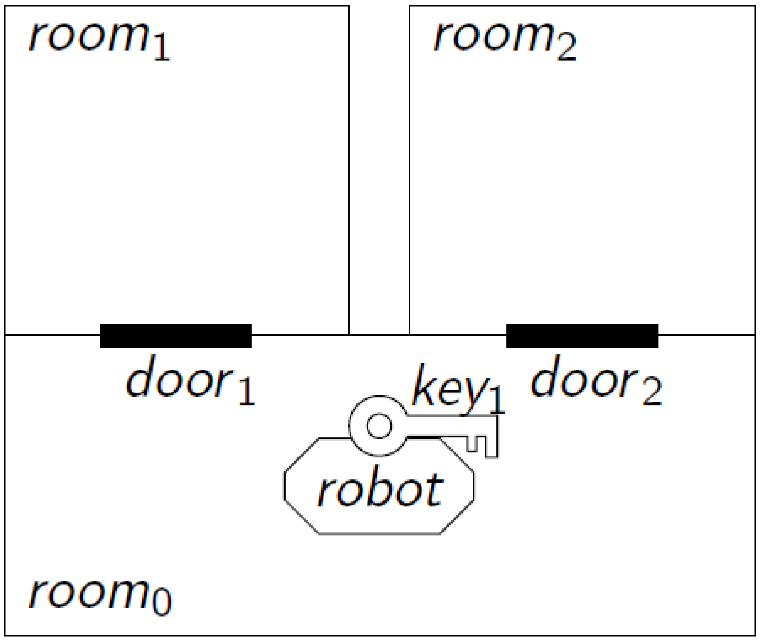
An excuse is a changed initial state in which:

- The values of fluents can be changed
- New objects can be added

Exclude those changes that

- make goal atoms immediately true.
- change a fluent that contributes to the goal.

Example

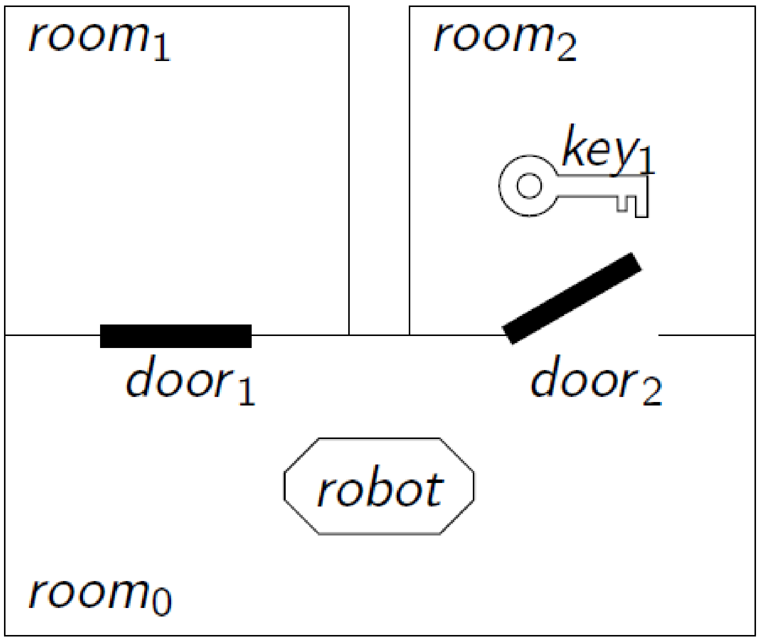


Possible excuses:

- $\{\mathrm{open}(door_1)\}$
- $\{\mathrm{open}(door_2)\}$
- $\{\mathrm{pos}(key_1) = robot\}$
- $\{\mathrm{pos}(key_1) = room_0\}$

Why did you not open the door yourself?
Because I do not have the key.

Example

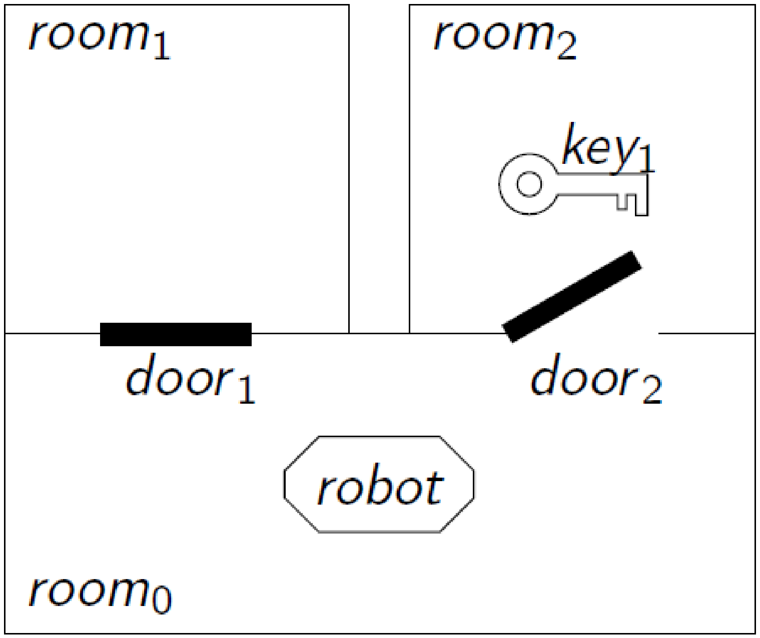


Possible excuses:

- $\{\text{open}(door_1)\}$
- $\{\text{open}(door_2)\}$
- $\{\text{pos}(key_1) = robot\}$
- $\{\text{pos}(key_1) = room_0\}$

Why did you not get the key yourself?
Because $door_2$ is not open.

Example



Possible excuses:

- $\{\text{open}(door_1)\}$
- $\{\text{open}(door_2)\}$
- $\{\text{pos}(key_1) = robot\}$
- $\{\text{pos}(key_1) = room_0\}$

Why did you not open $door_2$ yourself?
Because there is no way to open $door_2$.

Good Excuses

- An excuse that can be regressed to another excuse is no good excuse.
- Static facts are always good excuses.

Perfect Excuses

- Some facts about the world are “more static” than others.
  - Example: adding a new key vs. adding a new door.
- Associate costs with changing facts.
- A good excuse with minimal costs is a perfect excuse.

Finding excuses

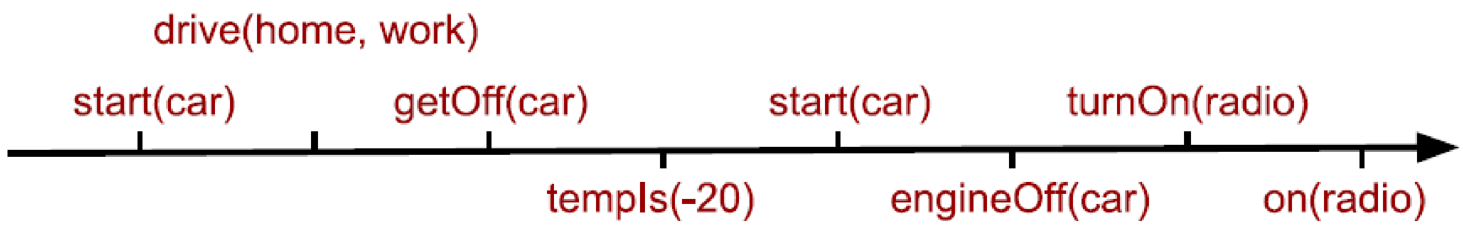
- Reduce the problem of finding excuses to planning.
- Introduce change operators that can modify the initial state at will.
- Partition the plan into two phases.
- Allow application only in the initial phase

Finding excuses

- For efficiency, limit the number of change operators.
- We limit ourselves to static facts and facts on cycles.
  - Determine cyclic facts using the ungrounded causal graph.
- Sufficient for some planning problems
  - In general, might not find all good excuses.

Explanations

Preferred Explanations
Sohrabi, Baier & McIlraith (2011)



What could explain this?

- Many things: battery died, leads wet, ran out of gas, ...

But not all explanations are equal...

E.g., *Preferences* can be expressed over explanations for the car not starting:

- If radio is dead then sometime in the past the battery died.
- If it is rainy then sometime in the past high tension leads got wet.

OVERVIEW

1. INTRODUCTION
2. INTERPRETATION
3. DECISION SUPPORT
    a. Explicit Constraints
    b. Implicit Constraints (Preferences)
    c. Incomplete Dynamics
4. COMMUNICATION
    a. Excuses & Explanations
    b. Asking for Help
5. CASE STUDY
6. SUMMARY

Planning with Human Help

Modeling humans as actors along with robots is computationally intractable for large numbers of humans

Instead, model humans only as observation/actuation providers to the robot with limited availability and accuracy

How to Ask for Help

Symbiotic Autonomy
Rosenthal, Veloso et al.

Can you point to where we are on this map?

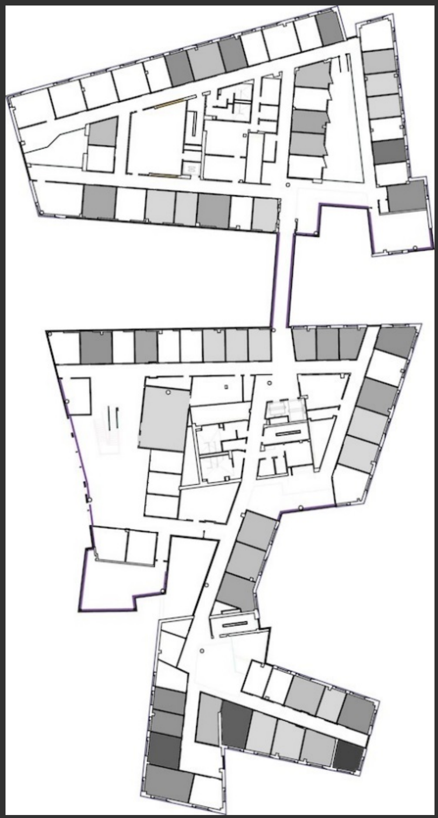
People often give grounding context when asking for help

[Rosenthal, Veloso ,Dey: Ro-Man 2009, IUI 2010, JSORO 2012]

Who To Ask: Results

Environment Occupants are not Always Available

Availability of Help



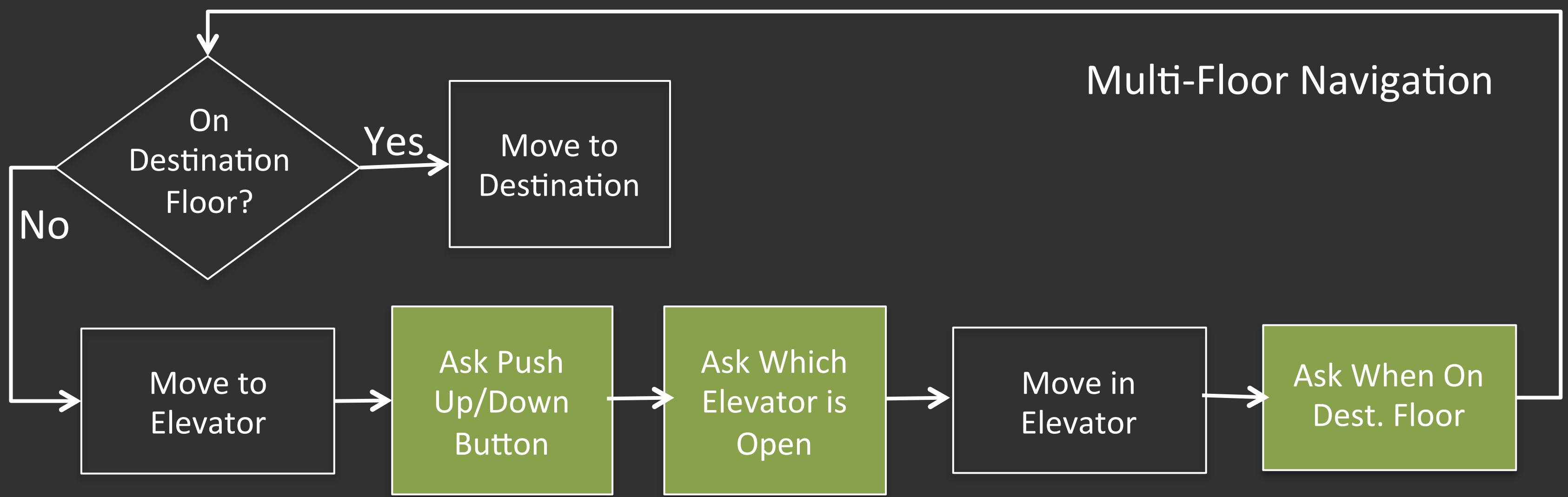
Shortest path may not be the one with the most available or willing help

Actuation Limitations

Asking for Help Assuming Humans Always Available

Ask Push Up/Down Button → Ask Which Elevator is Open → Move in Elevator → Ask When On Dest. Floor

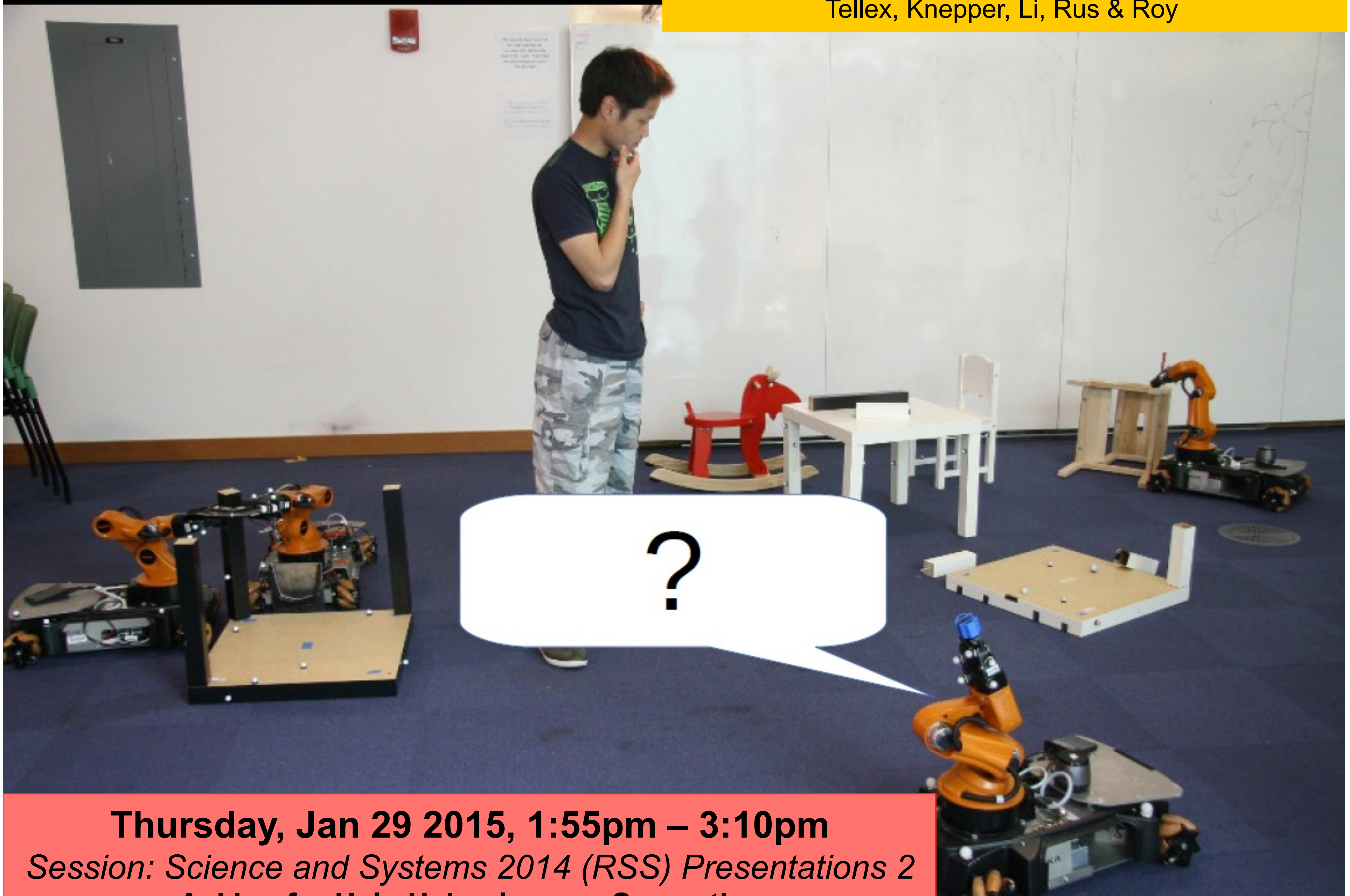
Conditional Plan to Overcome Actuation Limitations



Enable New Functionality by Requesting Help

Asking for Help Using Inverse Semantics

Tellex, Knepper, Li, Rus & Roy

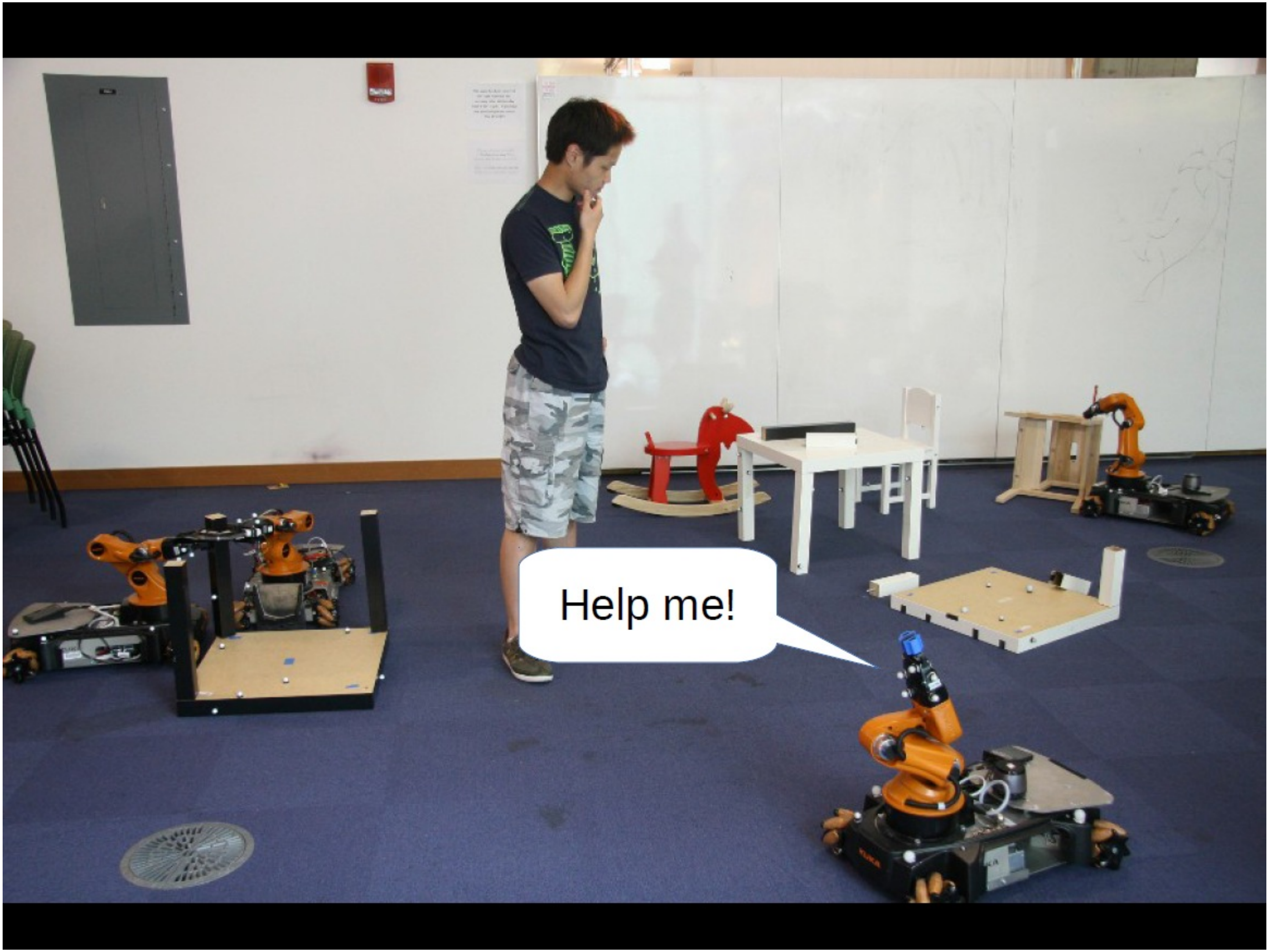


Thursday, Jan 29 2015, 1:55pm – 3:10pm

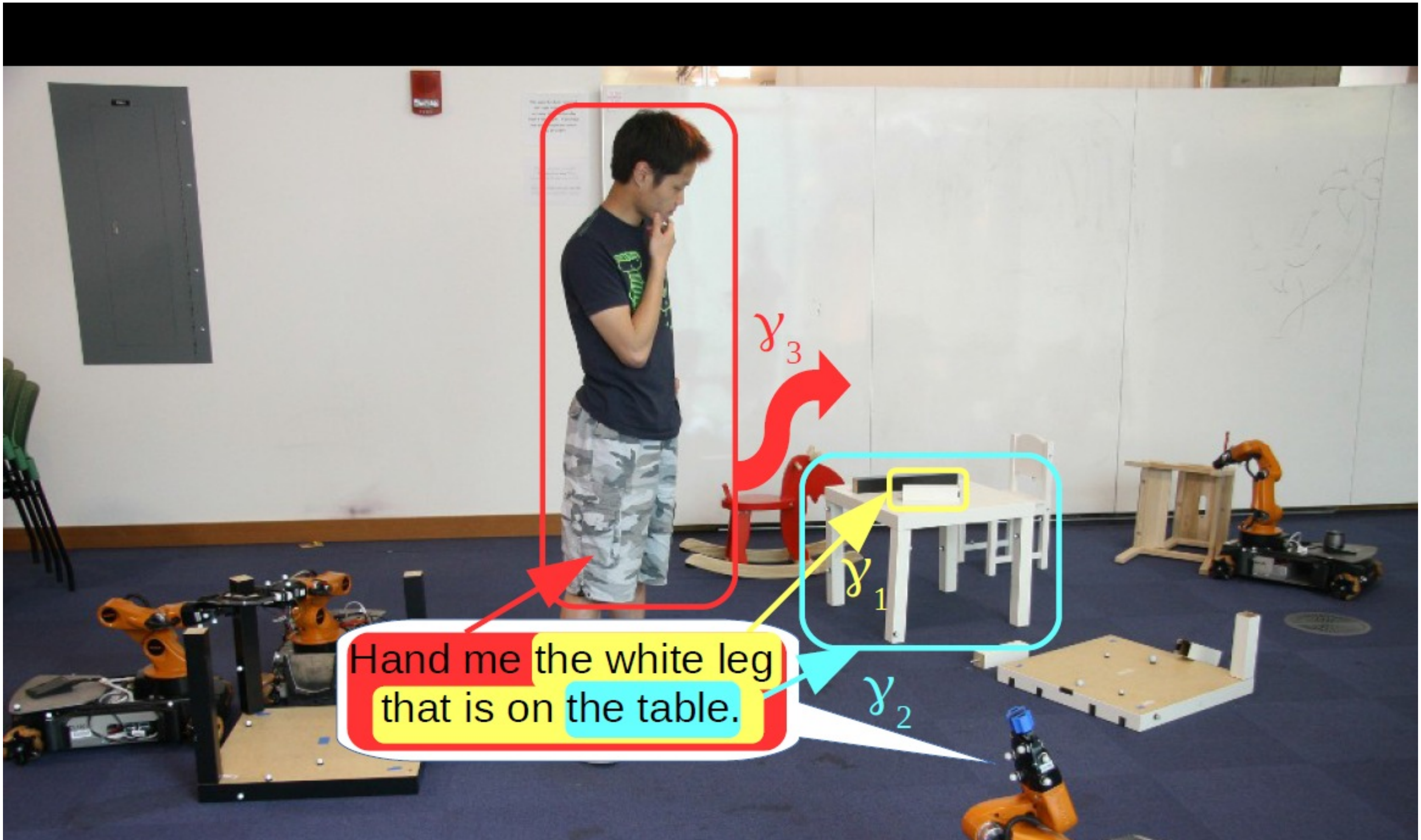
Session: Science and Systems 2014 (RSS) Presentations 2

Asking for Help Using Inverse Semantics

Stefanie Tellex, Ross Knepper, Adrian Li, Daniela Rus, Nicholas Roy


Help me!


Please hand me the white table leg.



γ_k are *groundings*, or objects, places, paths, and events in the external world. Each γ_k corresponds to a constituent phrase in the language.

OVERVIEW

1. INTRODUCTION
2. INTERPRETATION
3. DECISION SUPPORT
   a. Explicit Constraints
   b. Implicit Constraints (Preferences)
   c. Incomplete Dynamics
4. COMMUNICATION
   a. Excuses & Explanations
   b. Asking for Help
5. CASE STUDY
6. SUMMARY

Case Study: Planning for Human-Robot Teaming

- **Human-Robot Teaming (HRT)** is becoming an important problem

- Requires a lot of different technologies
  - Perception (Vision), Actuation, Dialogue, **Planning** …

- Most current **robots** are **glorified remote-operated sensors**

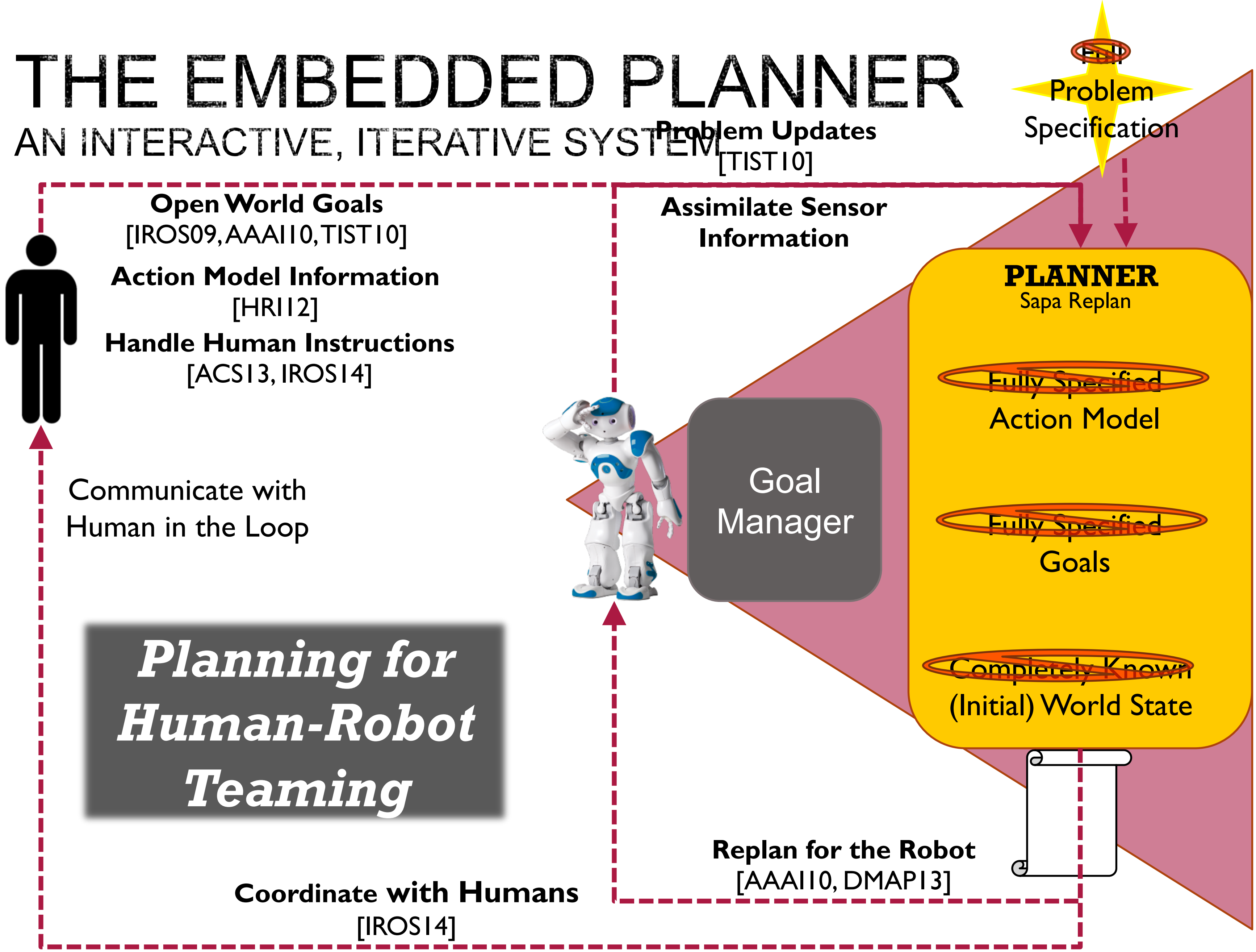
- Autonomous Planning is an important capability
  - Supporting ***flexible*** **HRT** with **constant changes**

Planning Challenges in Human-Robot Teaming

1. **OPEN WORLD GOALS**
   - Provide a way to specify quantified goals on unknown objects
   - Consider a more principled way of handling uncertainty in facts
2. **REPLANNING**
   - Handle state and goal updates from a changing world while executing
   - Present a unified theory of replanning, to analyze tradeoffs
3. **MODEL UPDATES**
   - Accept changes to planner's domain model via natural language
4. **PLAN RECOGNITION**
   - Use belief models of other agents to enhance planning

THE EMBEDDED PLANNER

AN INTERACTIVE, ITERATIVE SYSTEM



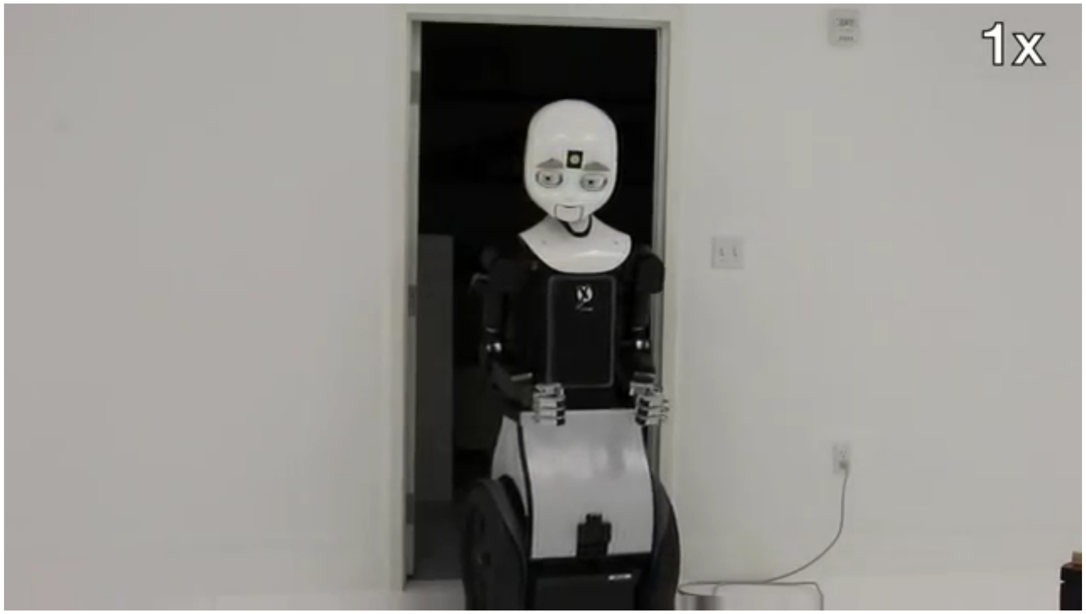
Fielded Prototype

- Planning Artifact: **Sapa Replan**
  - Extension of **Sapa** metric temporal planner
- **Partial Satisfaction Planning**
  - Builds on Sapa^{PS} planner
- **Replanning**
  - Uses an execution monitor to support scenarios with real-time execution

[Benton et al., AIJ07]
[Talamadupula, Benton, et al., TIST10]

Open World Goals

- When to start sensing?
  - Indicator to start sensing
- What to look for?
  - Object type
  - Object properties



- When to stop sensing?
  - When does the planner know the world is closed?
- Why should the robot sense?
  - Does the object fulfill a goal?
  - What is the reward? Is it a bonus?

[Talamadupula, Benton et al., ACM TIST 2010]

Replanning for Changing Worlds

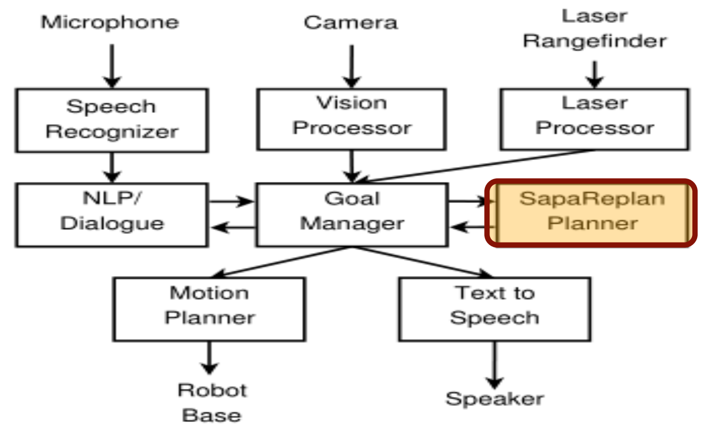
- New Information
  - Sensors
  - Human teammate

- New Goals
  - Orders: Humans
  - Requests
- Requirement
  - New plan that works in new world (state)
  - Achieves the changed goals

[Talamadupula et al. AAAI10]

Model Updates
(via natural language)

- "To go into a room when you are at a closed door, push it one meter."
  - Precondition: "you are at a closed door"
  - Action definition: "push it one meter"
  - Effect: "go into a room"



- NLP Module
  i. Reference resolution
  ii. Parsing
  iii. Background knowledge
  iv. Action submission (to planner)

[Cantrell, Talamadupula et al., HRI 2012]

[In collaboration with hrilab, Tufts University]

Example: Action Addition

New Action: "push"

"To go into a room when you are at a closed door, push it one meter."

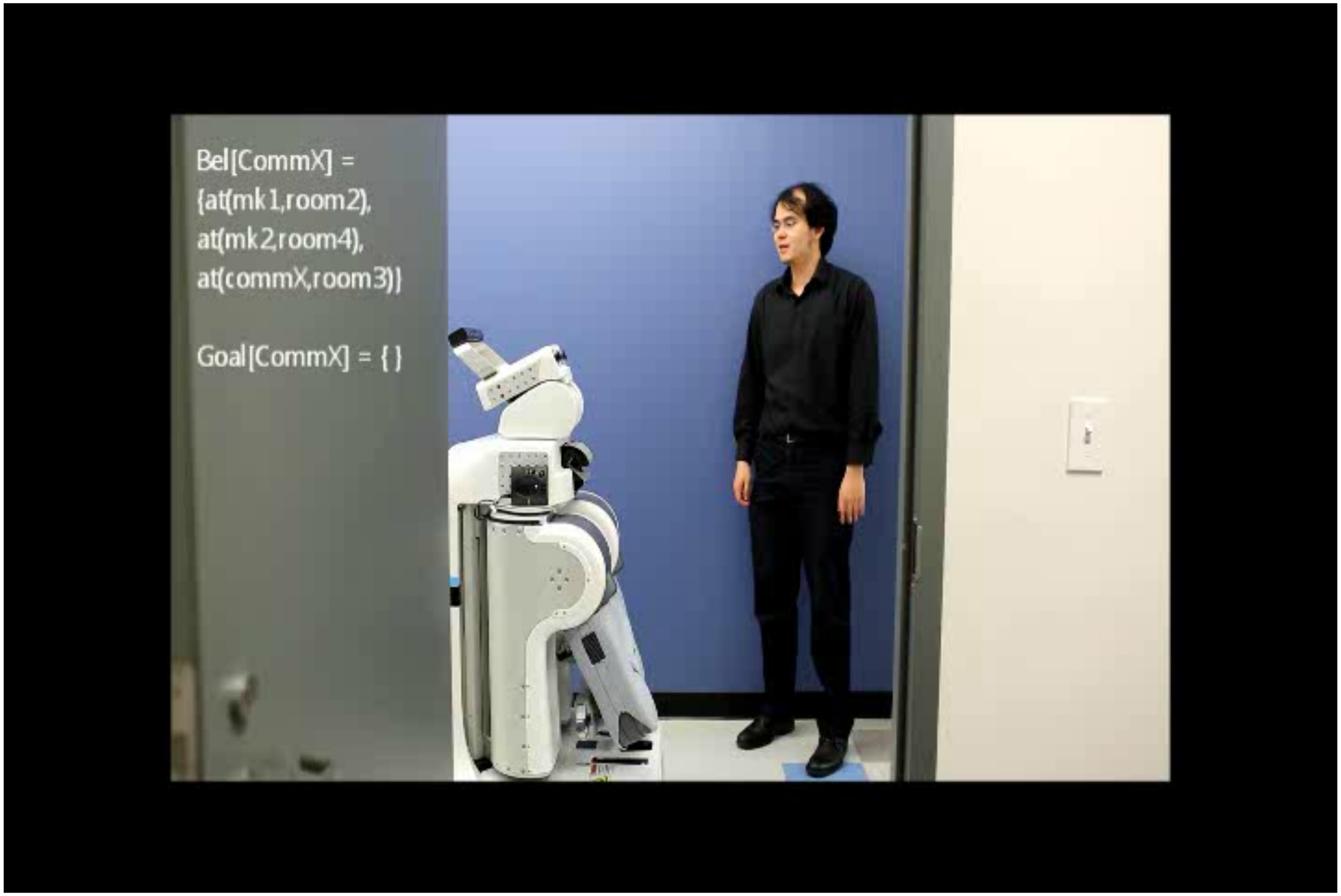
```
(:durative-action push
 :parameters (?door - doorway ?cur_loc - hallway ?to_loc - zone)
 :duration (= ?duration (dur_push))
 :condition (and (at start (at ?cur_loc))
                 (at start (door_connected ?door ?cur_loc ?to_loc))
                 (over all (door_connected ?door ?cur_loc ?to_loc)))
 :effect (and (at start (not (at ?cur_loc)))
              (at end (open ?doorway))
              (at end (at ?to_loc))))
```

From natural language

Architecture

Background knowledge

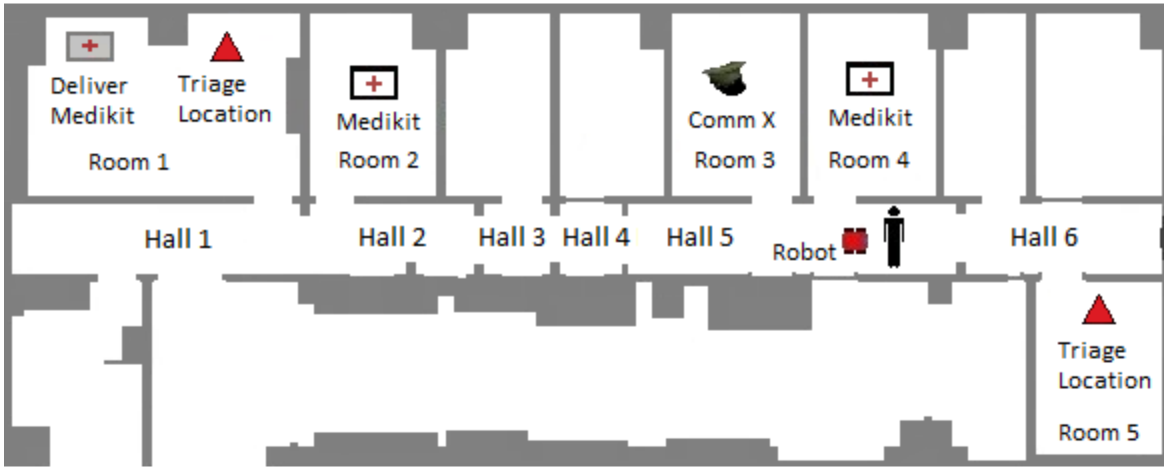
Plan & Intent Recognition



[In collaboration with hrilab, Tufts University]

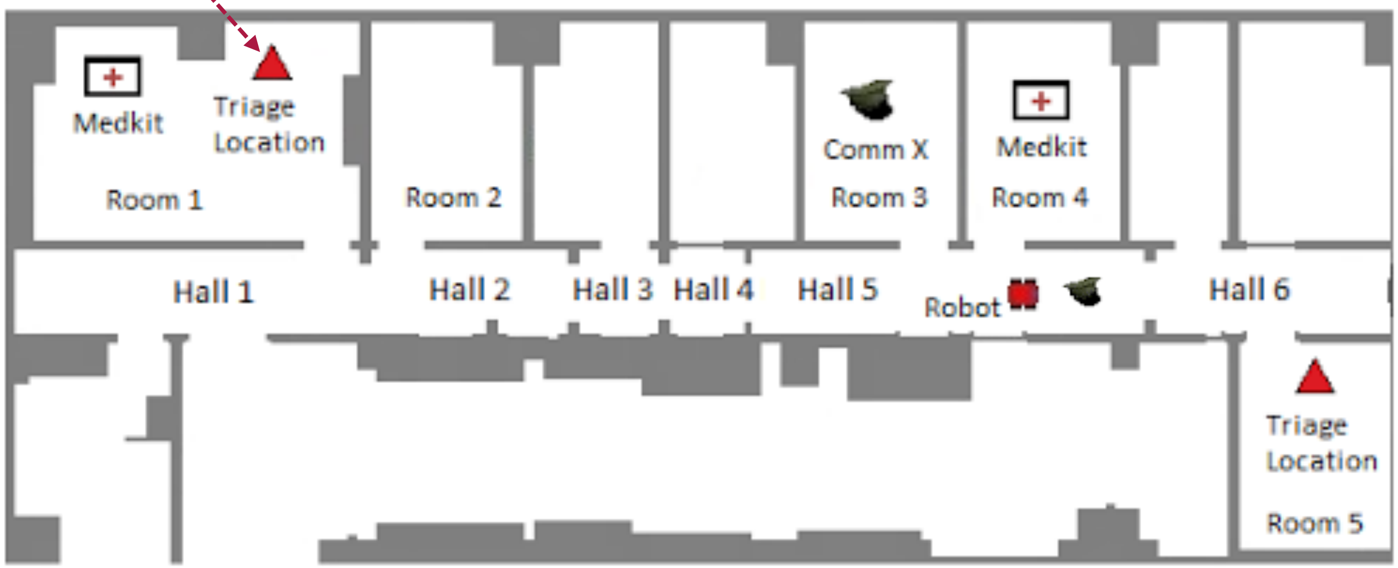
[Talamadupula, Briggs et al., IROS14]

Plan & Intent Recognition



1. Map the robot's beliefs and knowledge about CommX into a new planning instance
2. Generate a plan for this instance – prediction of CommX's plan
3. Extract relevant information from the predicted plan
   - Which medkit will CommX pick up?
4. Use the extracted information to deconflict robot's plan

Comm X's Goal



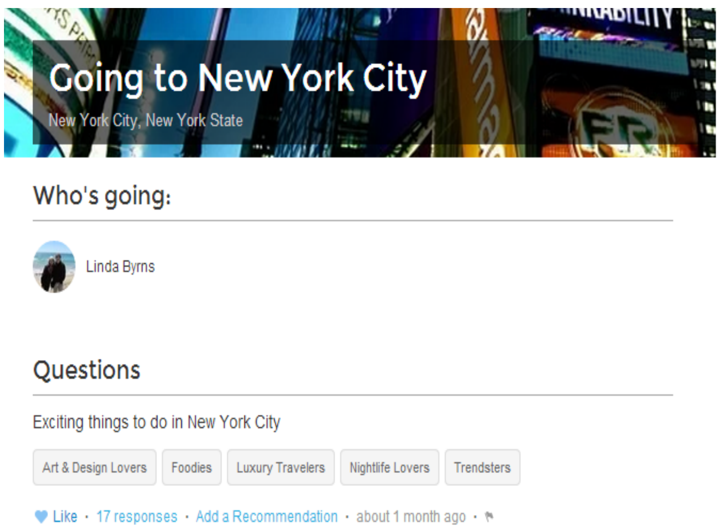
PREDICTED PLAN FOR COMMX

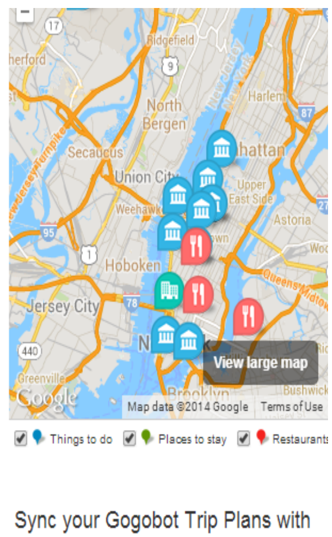
move commx room3 hall5
move_reverse commx hall5 hall4
move_reverse commx hall4 hall3
move_reverse commx hall3 hall2
move_reverse commx hall2 hall1
move_reverse commx hall1 room1
pick_up_medkit commx mkeast room1
conduct_triage commx room1

Case Study: Crowd-Planning

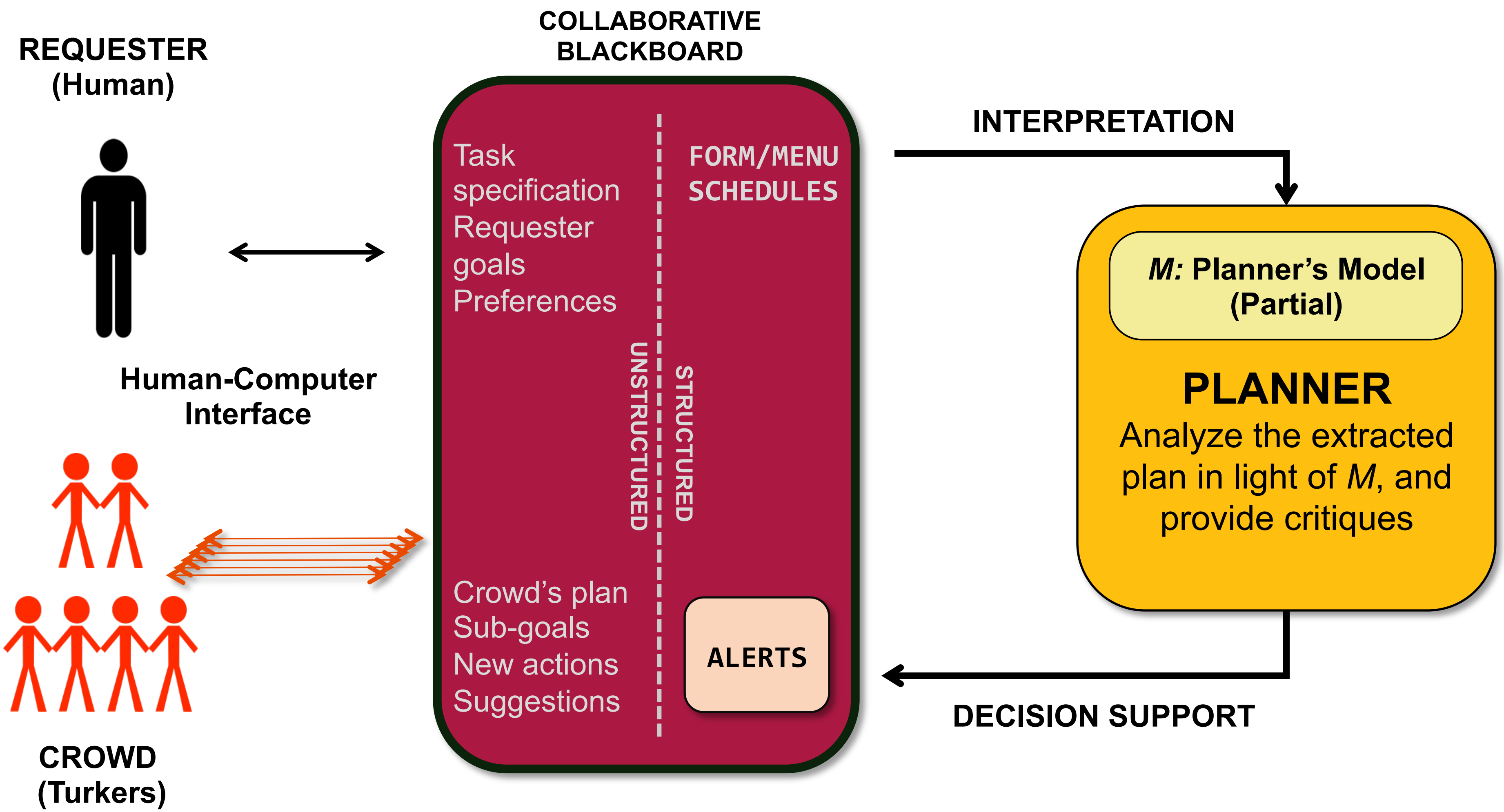
- Crowdsourcing: Process of obtaining ideas or a needed service from a crowd of people
  - Crowd + OutSourcing

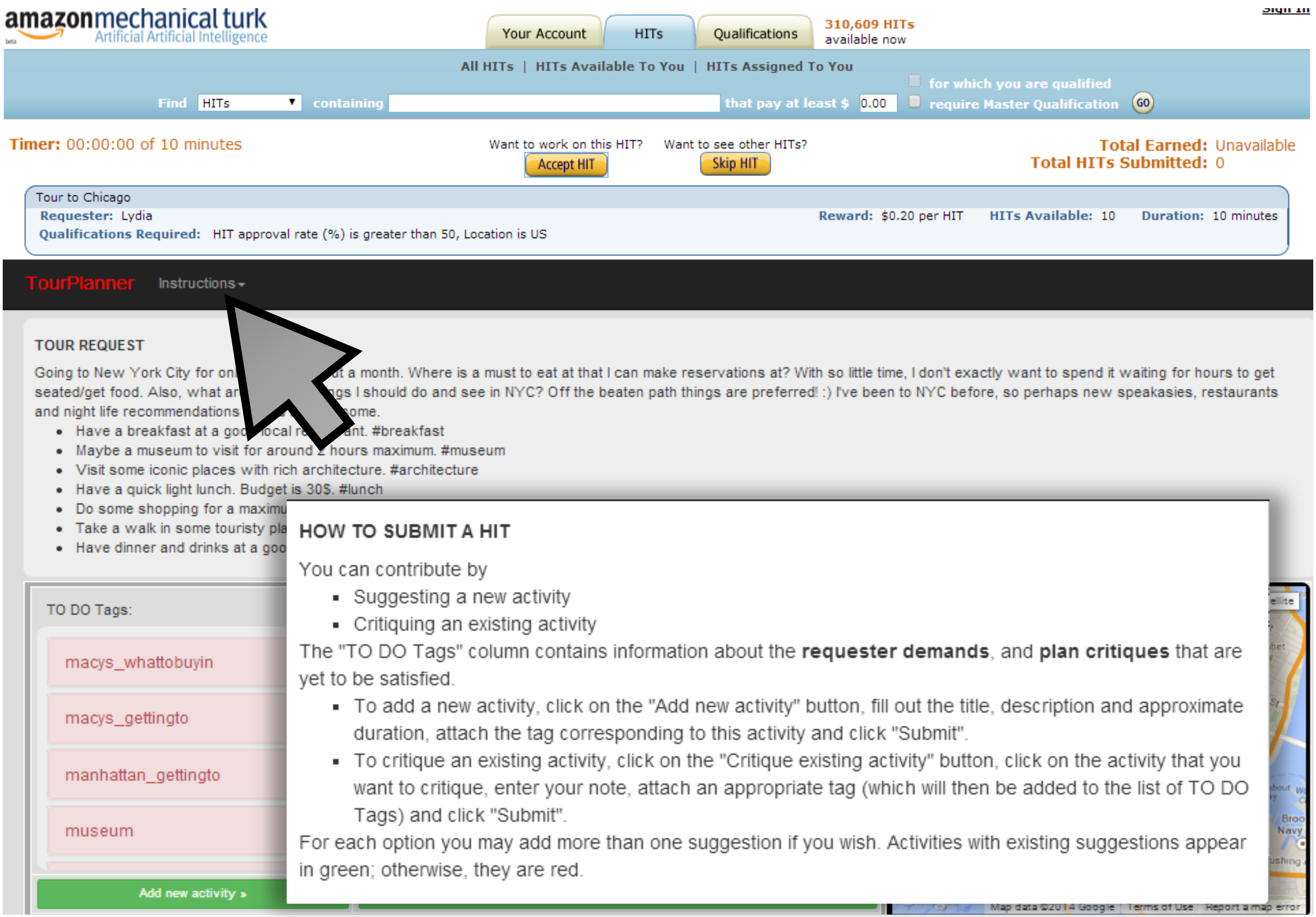
- Example: Travel Planning





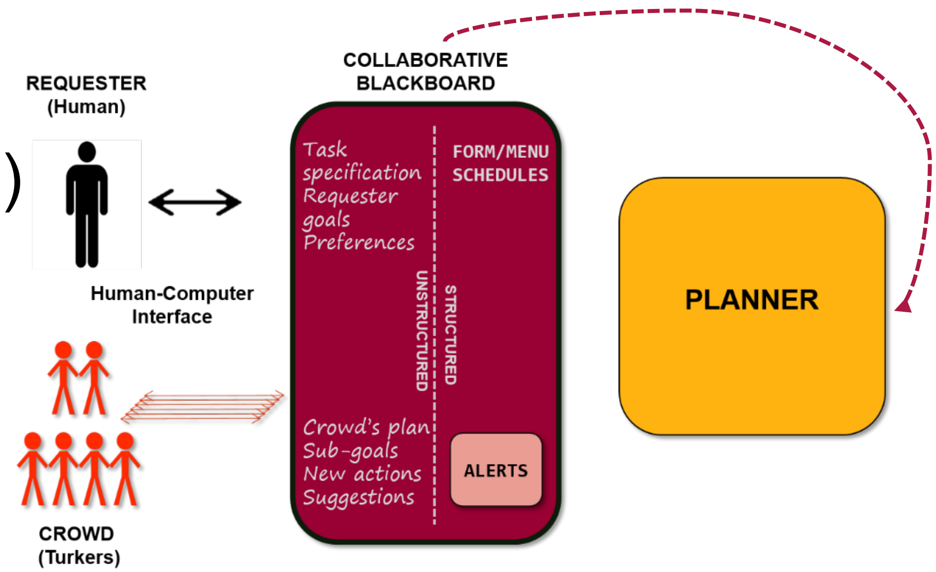
AI-MIX: A CROWD-PLANNING SYSTEM




amazonmechanical turk
Artificial Artificial Intelligence
Your Account
HITs
Qualifications
310,609 HITs
available now
All HITs | HITs Available To You | HITs Assigned To You
for which you are qualified
Find HITs containing that pay at least $ 0.00 require Master Qualification GO
Timer: 00:00:00 of 10 minutes
Want to work on this HIT? Accept HIT
Want to see other HITs? Skip HIT
Total Earned: Unavailable
Total HITs Submitted: 0
Tour to Chicago
Requester: Lydia
Reward: $0.20 per HIT
HITs Available: 10
Duration: 10 minutes
Qualifications Required: HIT approval rate (%) is greater than 50, Location is US
TourPlanner
Instructions
TOUR REQUEST
Have a breakfast at a good local restaurant. #breakfast
Maybe a museum to visit for around 2 hours maximum. #museum
Visit some iconic places with rich architecture. #architecture
Have a quick light lunch. Budget is 30$. #lunch
TO DO Tags:
macys_whattobuyin
macys_gettingto
manhattan_gettingto
museum
Add new activity »
HOW TO SUBMIT A HIT
You can contribute by
Suggesting a new activity
Critiquing an existing activity
The "TO DO Tags" column contains information about the requester demands, and plan critiques that are yet to be satisfied.
To add a new activity, click on the "Add new activity" button, fill out the title, description and approximate duration, attach the tag corresponding to this activity and click "Submit".
To critique an existing activity, click on the "Critique existing activity" button, click on the activity that you want to critique, enter your note, attach an appropriate tag (which will then be added to the list of TO DO Tags) and click "Submit".
For each option you may add more than one suggestion if you wish. Activities with existing suggestions appear in green; otherwise, they are red.
Map data ©2014 Google
Terms of Use
Report a map error

CHALLENGE: INTERPRETATION

- Understanding the goals and plans of the humans (requester + crowd) from semi-structured or unstructured text
- Impedance Mismatch



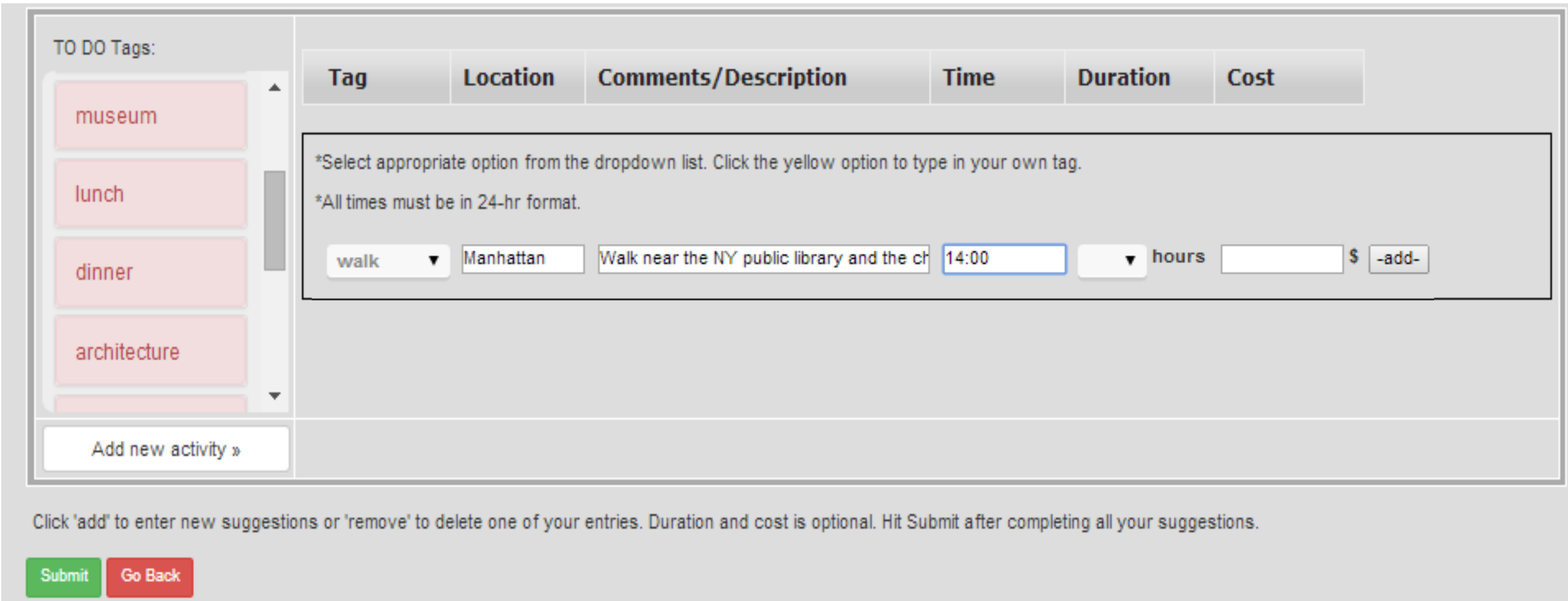
Extract from Plain Text
Impose structure
[Ling & Weld, 2010]
[Kim, Chacha & Shah, 2013]

Full Plan Recognition
[Kautz & Allen, 1986]
[Ramirez & Geffner, 2010]

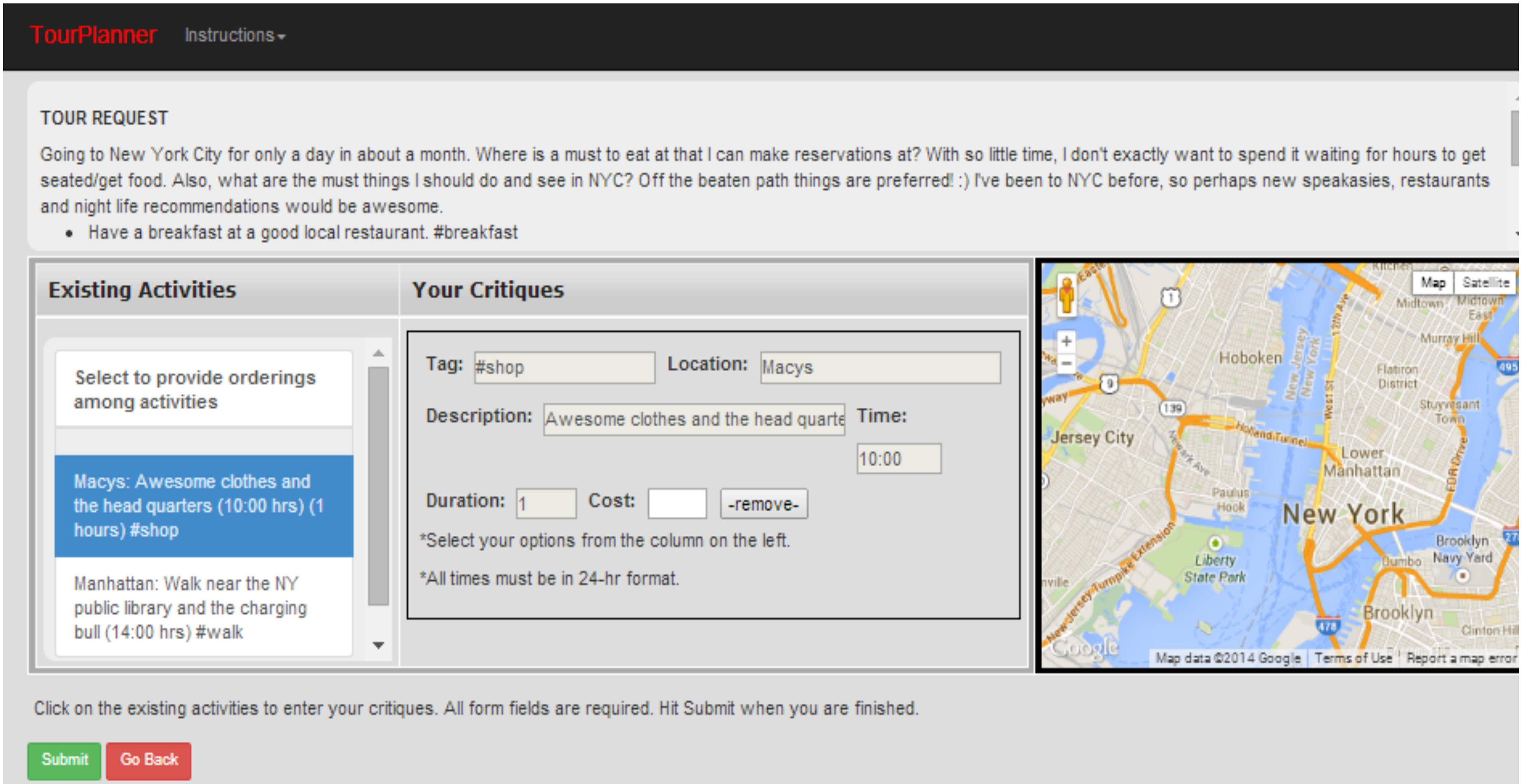
UNSTRUCTURED — **STRUCTURED**

Plan Recognition from Noisy Traces
Extract noisy traces first
[Zhuo, Yang & Kambhampati, 2012]

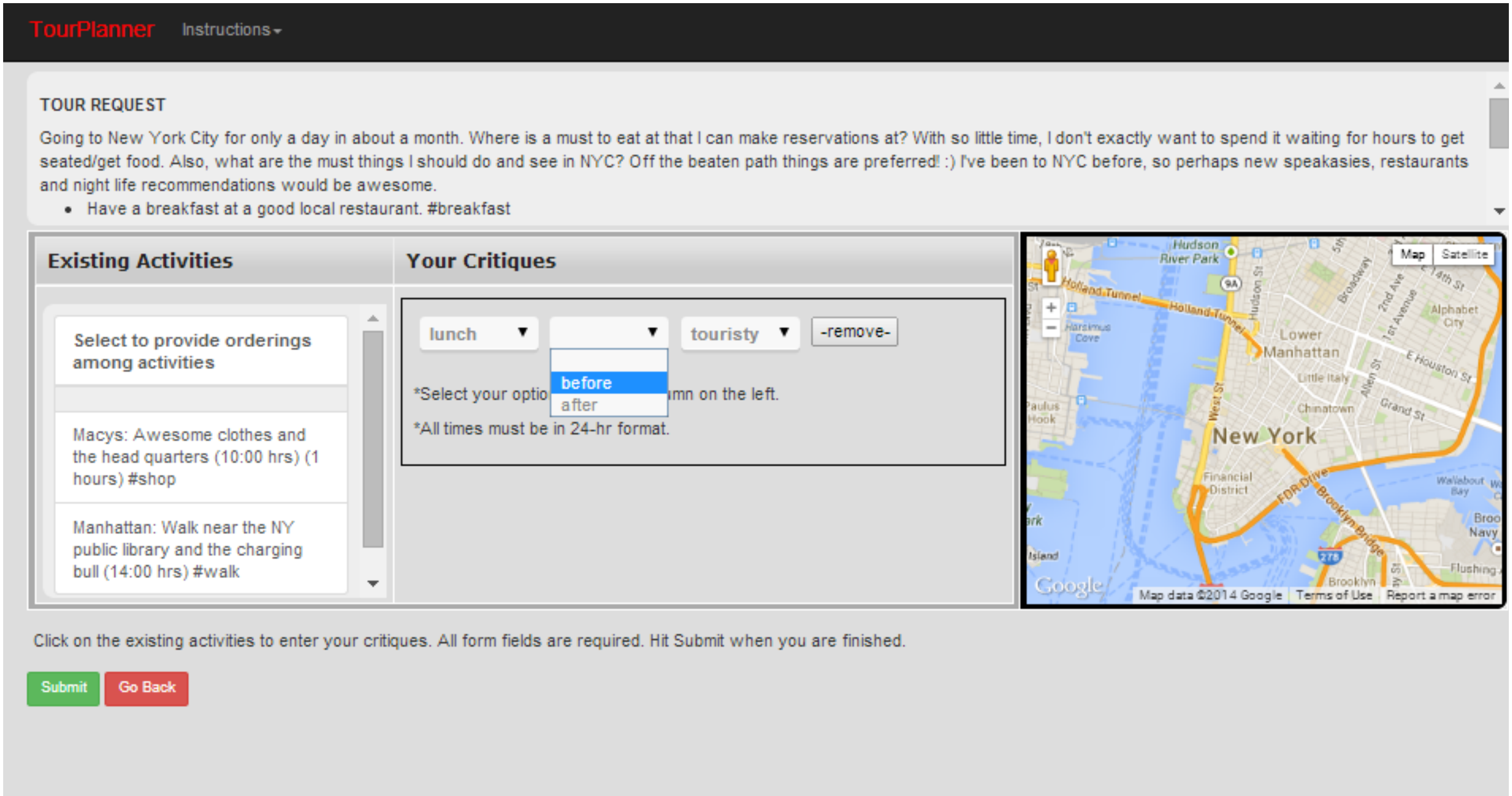
Add a New Suggestion



ADDING A CRITIQUE

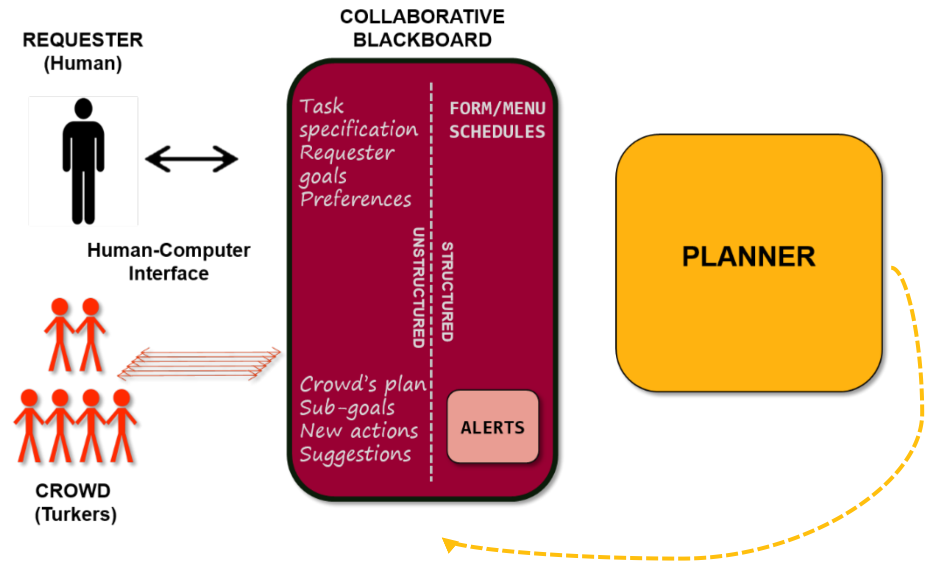


ADDING AN ORDERING CONSTRAINT



CHALLENGE: DECISION SUPPORT

- Steering the crowd workers towards producing a plan collaboratively
  - Partial domain dynamics
  - Incomplete preferences
- Iterative Process



Problem Decomposition
[Nau et al., 2003]

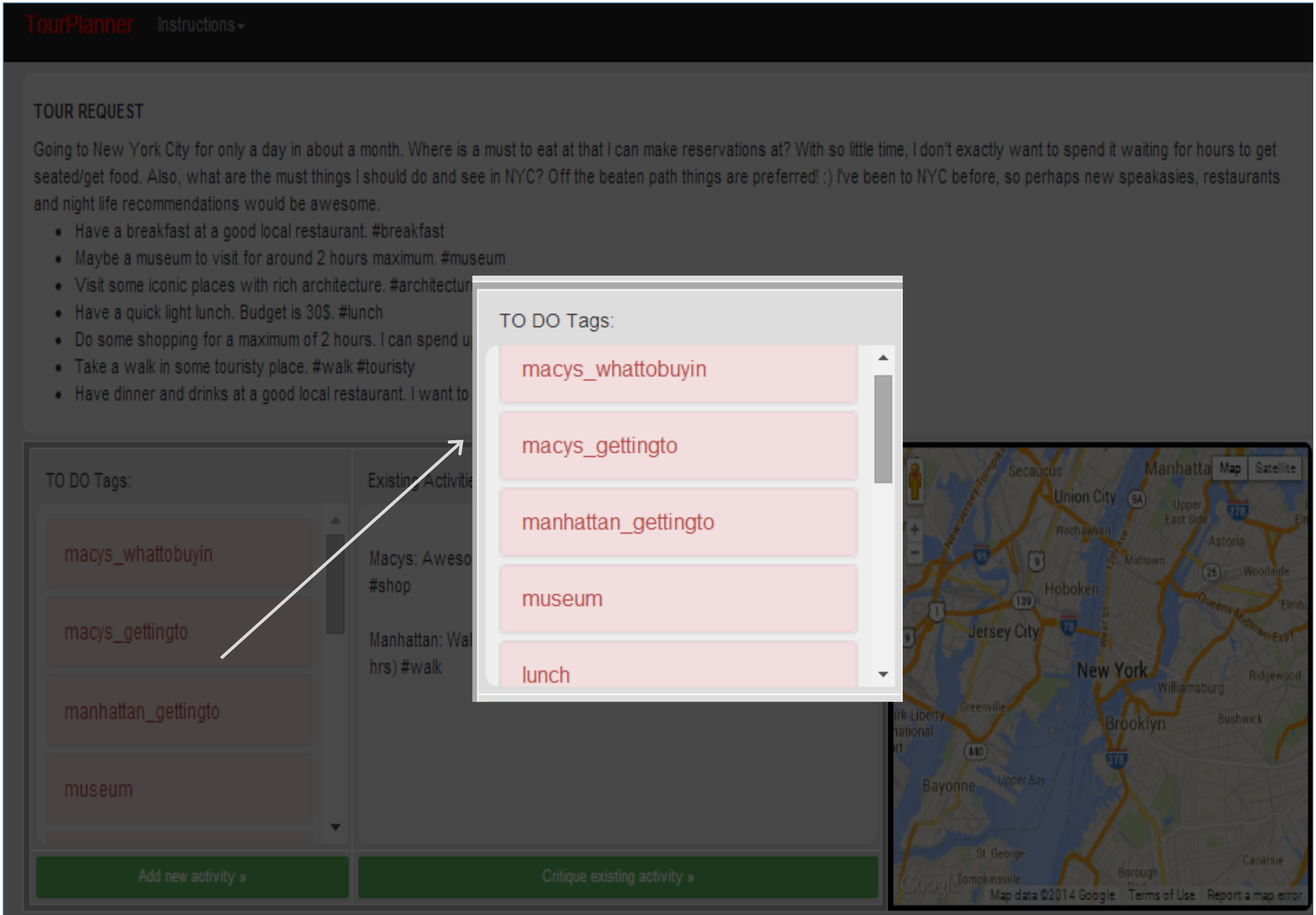
Plan Critiquing

Plan Generation

SHALLOW MODEL — **APPROXIMATE MODEL**

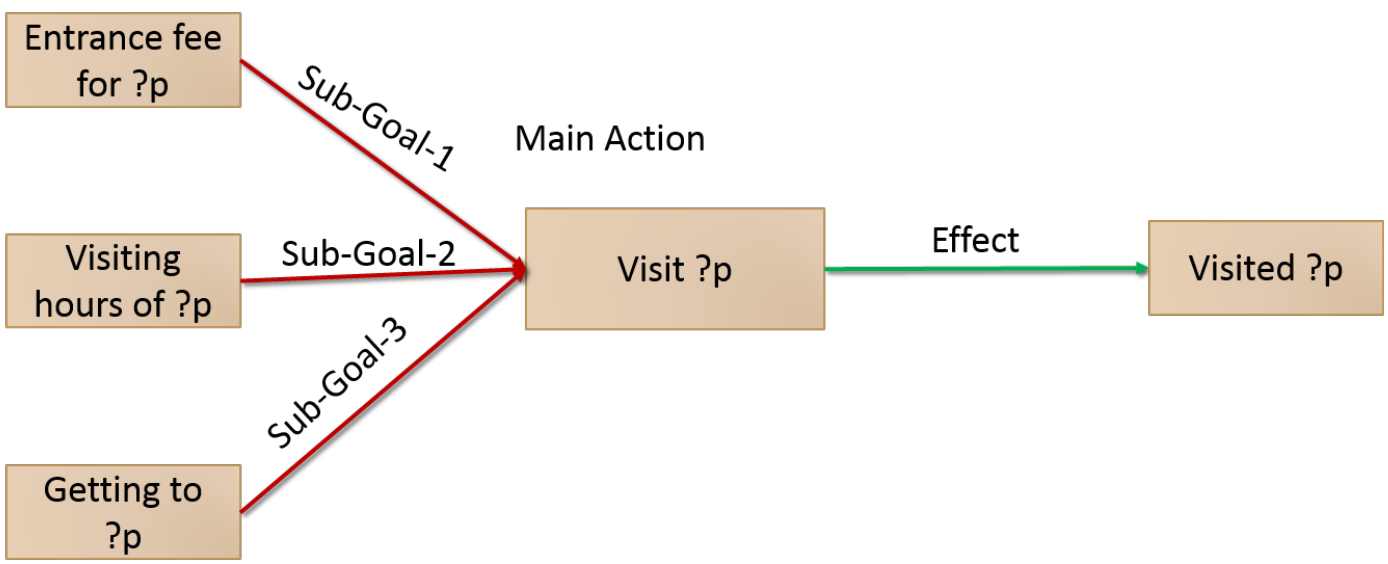
Constraint & Arithmetic Checking
Mobi [Zhang et al., 2012]
Cobi [Zhang et al., 2013]

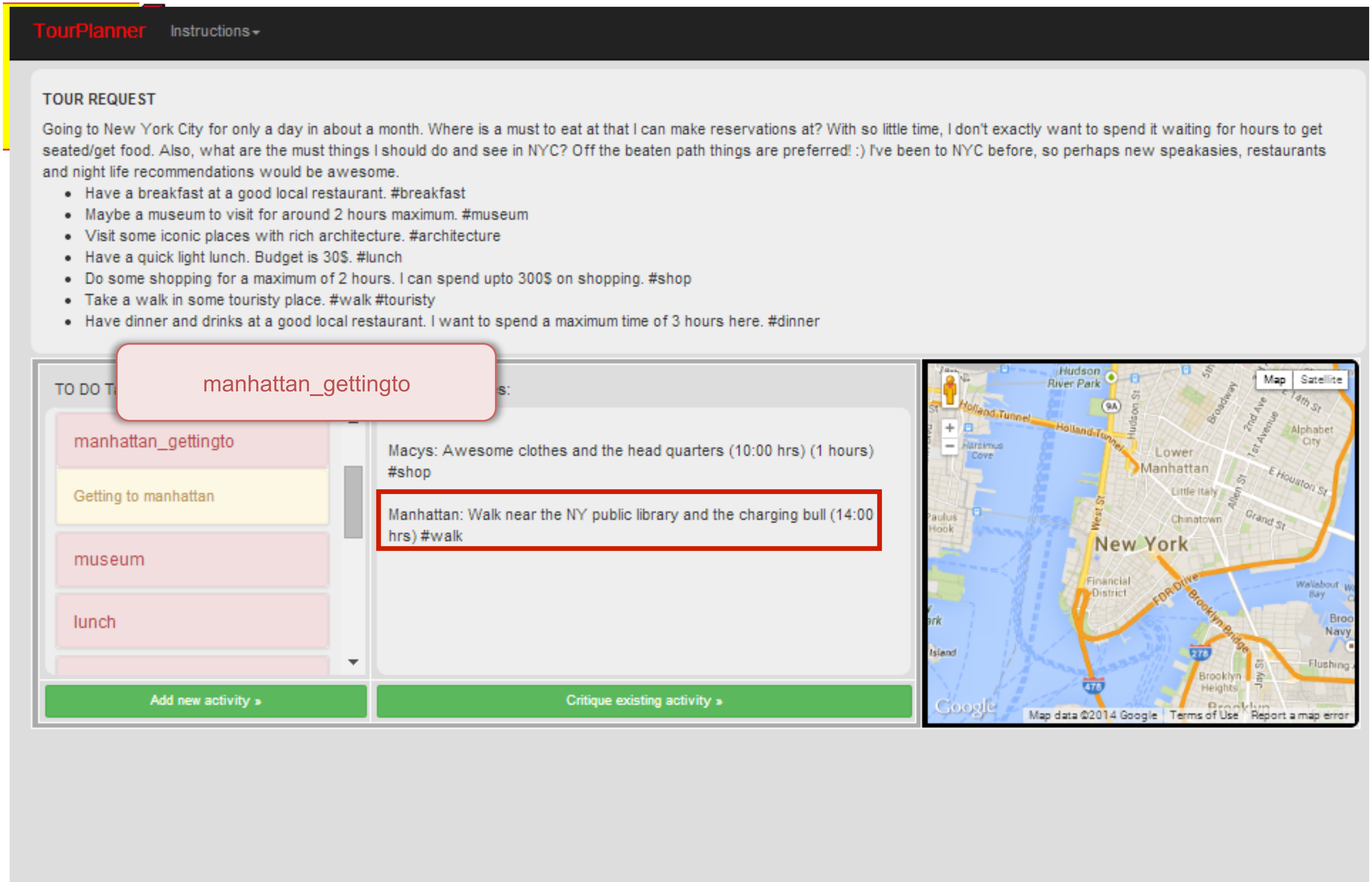
Regression
Match sub-goals with actions


TourPlanner
Instructions
TOUR REQUEST
Going to New York City for only a day in about a month. Where is a must to eat at that I can make reservations at? With so little time, I don't exactly want to spend it waiting for hours to get seated/get food. Also, what are the must things I should do and see in NYC? Off the beaten path things are preferred! :) I've been to NYC before, so perhaps new speakasies, restaurants and night life recommendations would be awesome.
Have a breakfast at a good local restaurant. #breakfast
Maybe a museum to visit for around 2 hours maximum. #museum
Visit some iconic places with rich architecture. #architectur
Have a quick light lunch. Budget is 30$. #lunch
Do some shopping for a maximum of 2 hours. I can spend u
Take a walk in some touristy place. #walk #touristy
Have dinner and drinks at a good local restaurant. I want to
TO DO Tags:
macys_whattobuyin
macys_gettingto
manhattan_gettingto
museum
lunch
Existing Activitie
Macys: Aweso
#shop
Manhattan: Wal
hrs) #walk
Add new activity »
Critique existing activity »
Map
Satellite
Secaucus
Union City
Hoboken
Jersey City
New York
Brooklyn
Bayonne
Williamsburg
Astoria
Map data ©2014 Google
Terms of Use
Report a map error

SUB-GOAL GENERATION
DECISION SUPPORT & COMMUNICATION

- Planner uses a high level PDDL action model
- Action examples: visit, lunch, shop …
- Generic preconditions
- Unsatisfied sub-goals thrown as alerts




TourPlanner
Instructions
TOUR REQUEST
Going to New York City for only a day in about a month. Where is a must to eat at that I can make reservations at? With so little time, I don't exactly want to spend it waiting for hours to get seated/get food. Also, what are the must things I should do and see in NYC? Off the beaten path things are preferred! :) I've been to NYC before, so perhaps new speakasies, restaurants and night life recommendations would be awesome.
Have a breakfast at a good local restaurant. #breakfast
Maybe a museum to visit for around 2 hours maximum. #museum
Visit some iconic places with rich architecture. #architecture
Have a quick light lunch. Budget is 30$. #lunch
Do some shopping for a maximum of 2 hours. I can spend upto 300$ on shopping. #shop
Take a walk in some touristy place. #walk #touristy
Have dinner and drinks at a good local restaurant. I want to spend a maximum time of 3 hours here. #dinner
manhattan_gettingto
TO DO T
manhattan_gettingto
Getting to manhattan
museum
lunch
Macys: Awesome clothes and the head quarters (10:00 hrs) (1 hours) #shop
Manhattan: Walk near the NY public library and the charging bull (14:00 hrs) #walk
Add new activity »
Critique existing activity »
Map
Satellite
Lower Manhattan
New York
Chinatown
Little Italy
Financial District
Brooklyn Heights
Map data ©2014 Google
Terms of Use
Report a map error

CASE STUDY: HRT + CROWD-PLANNING

	PLANNING FOR HUMAN-ROBOT TEAMING	PLANNING FOR CROWDSOURCING
INTERPRETATION	Open World Goals Continual (Re)Planning Plan & Intent Recognition	Activity Suggestions Activity Critiques Ordering Constraints
DECISION SUPPORT	Continual (Re)Planning Plan & Intent Recognition	Sub-goal Generation Constraint Violations Continual Improvement
COMMUNICATION	Model Updates	Sub-goal Generation

OVERVIEW

1. INTRODUCTION
2. INTERPRETATION
3. DECISION SUPPORT
   a. Explicit Constraints
   b. Implicit Constraints (Preferences)
   c. Incomplete Dynamics
4. COMMUNICATION
   a. Excuses & Explanations
   b. Asking for Help
5. CASE STUDY
6. SUMMARY

Recap

- HILP raises several open **challenges** for planning systems, depending on the modality of interaction between human and the planner
- **Contributions** of this tutorial:
  – Surveyed HILP scenarios
  – Discussed the dimensions along which they vary
  – Identified the planning challenges posed by HILP scenarios
    - Interpretation, Decision Support and Communication
  – Outlined current approaches for addressing these challenges
  – Presented detailed case-studies of two recent HILP systems (from our group ☺)

Dimensions of HIL Planning

	Cooperation Modality	Communication Modality	What is Communicated	Knowledge Level
Crowdsourcing	Interaction (Advice from planner to humans)	Custom Interface	Critiques, subgoals	Incomplete Preferences Incomplete Dynamics
Human-Robot Teaming	Teaming/ Collaboration	Natural Language Speech	Goals, Tasks, Model information	Incomplete Preferences Incomplete Dynamics (Open World)
"Grandpa Hates Robots"	Awareness (pre-specified constraints)	Prespecified (Safety / Interaction Constraints)	No explicit communication	Incomplete Preferences Complete Dynamics
MAPGEN	Interaction (Planner takes binding advice from human)	Direct Modification of Plans	Direct modifications, decision alternatives	Incomplete Preferences Complete Dynamics

Challenges for the Planner

- Interpret what humans are doing
  - Plan/goal/intent recognition
- Decision Support
  - Continual planning/Replanning
    - Commitment sensitive to ensure coherent interaction
    - Handle constraints on plan
  - Plan with incompleteness
    - Incomplete Preferences
    - Incomplete domain models
      - Robust planning with “lite” models
      - (Learn to improve domain models)
- Communication
  - Explanations/Excuses
    - Excuse generation can be modeled as the (conjugate of) planning problem
  - Asking for help/elaboration
    - Reason about the information value

Eigen Slide

(Other Relevant) Challenges (that are out-of-scope of this tutorial)

- Human Factors
  - How to make planning support "acceptable" to the humans in the loop?
  - How to adjust the planner autonomy to defer to the humans in the loop?
- Speech and Natural Language Processing in Collaborative Scenarios
- Learning to Improve models
  - Learning from demonstrations..
- Advances in multi-agent planning
  - Problem decomposition; Coordination etc.

Human-in-the-Loop Planning is making inroads...

- Several papers that handle these challenges of Human-Aware Planning have been presented at recent AAAI conferences (and ICAPS, IJCAI, IAAI…)
  - Significant help from applications tracks, robotics tracks and demonstration tracks
  - Several planning-related papers in non-ICAPS venues (e.g. AAMAS and even CHI) have more in common with the challenges of Human-aware planning
- ..so consider it for your embedded planning applications

going 

of *Planning*

Imagine there's no Landmarks
It's easy if you try
No benchmarks below us
Above us only blai
Imagine all the planners
Planning for real

Imagine there's no state
It isn't hard to do
Nothing to regress or relax
And no cost guidance too
Imagine all the planners
Lifting all the worlds

You may say that I'm a whiner
But I'm not the only one
I hope someday you'll join us
And the ICAPS will be more fun

Imagine there's no models
I wonder if you can
No need for preferences or groundings
A diversity of plans
Imagine all the planners
Living life incomplete

You may say that I'm a whiner
But I'm not the only one
I hope someday you'll join us
And the ICAPS will be more fun