Resolving Unachievable Goals through Collaborative Diagnosis

- Aero & Astro PhD Proposal Defense

Peng Yu
September 30th, 2014
• We are working with robots in industry and in our daily lives.
Challenges

• We don’t trust our robots:
  – **Communication barrier**: they do not understand plain English.
  – **Unreliable**: they often can’t do what they are told.
  – **Uncommunicative**: they do not communicate why they fail.
  – **Risky**: robots do not understand risk in real world.
Objective

• Enable robots to be **trustworthy teammates**.
  – Working with robots should be as easy and as safe as with humans.

• Trusted robots should embody four behaviors:
  – Communicate simply.
  – Explain causes.
  – Propose alternatives.
  – Sensitive to risk.

• Solution: Uhura executive.

• Focus: real-world task planning problems.
Outline

• Example Interaction
• Problem Statement
• Approach
• Progress
• Thesis Plan
Example: Wing Sub-assembly for Aircraft Manufacturing

- A technician is working with a team of industrial robots on the production line.
- His task is to assemble two wing boxes.

A Completed Wing Sub-assembly
(Simplified) Wing Assembly Procedures

• **Step 1: Lift and Align** aluminum skin with a wing box assembly.
  – This is done by a pair mobility platforms.

`Workers align aluminum skin with wing assembly`

`XWAMs align aluminum skin with wing assembly`
(Simplified) Wing Assembly Procedures

- **Step 2: Rivet** the skin to the wing box using Cleco fixture.
  - This is done by a riveting robot (Baxter).

Baxter... plans where it will cleco loads cleco into gun
Fastens skin to frame
An Unexpected Failure Just Occurred

• The assembly process is interrupted by the unexpected failure of the riveting robot.
  – The plan is no long feasible, since no other robot is capable of riveting wing boxes.

• The technician turns to his decision support system, Uhura, to discuss about the impact and options to repair the broken plan:
A Failure-Driven Dialog For Problem Resolution

I want to assemble two wingboxes by 7pm.

I cannot find a plan to complete our task.

Why? Because Riveting is necessary for making the wing box, but the remaining robots are incapable of riveting.

OK, what options do we have?

1 Communicates Simply

2 Explains causes
Alternatively, we can still complete the tasks for today if the riveting robot is back online by noon.

Can we have the riveting robot repaired in an hour?

No. The workshop is repairing it, but it will be offline for at least 12 hours.
OK, can you call in off-duty staff to help us?

If he arrives in 2 hours, there is 80% chance that we can complete the task by 8pm.

I can call in staff to help, but the wing assembly must be completed before 7pm.
Could you help rivet the wing boxes until the staff arrives?

You are qualified for this task. We can then complete the wing repair on time.

OK, lets go with that plan.
Recap of Objective

• Enable robots to be **trustworthy teammates.**
  – Working with robots should be as easy and as safe as with humans.

• Trusted robots should embody four behaviors:
  – Communicate simply.
  – Explain causes.
  – Propose alternatives.
  – Sensitive to risk.

• Solution: Uhura executive.

• Focus: real-world task planning problems.
Demonstration

- Demonstrate Uhura on:
  - Collaborative manufacturing tasks
  - Planning missions for deep-sea expeditions
Problem Statement

• Input to Uhura is a temporal planning problem\([5]\):
  – Goals;
  – Initial state;
  – Planning domain;
  – Objective function;
Time-evolved Goal Representation

- Goals are represented as Qualitative State Plans (QSPs), with three key elements:
  - **Episodes**: desired state trajectories.
  - **Temporal**: timing requirements between episodes.
  - **Resource** and **Chance** constraints: desired resource levels and risk bound over episodes.

Risk < 5%       Remaining Power > 50%
Planning Domain Specifications

- Extend PDDL descriptions ("STRIPS")

```
(:durative-action rivet
  :parameters (?r - rivet-robot  
    ?w - wingbox)
  :condition
    (and (at start (in-fixture ?w))
      (at start (available ?r)))
  :effect
    (and (at start (not (available ?r)))
      (at end (available ?r))
      (at end (assembled ?w))
      (at end (decrease (battery-level ?r) (normal 0.1 0.05)))
  :duration (= ?duration (normal 3.5 0.5)))
  :resource ())
```

- with

  - Simple temporal constraints (Simple Temporal Network).
  - Resource constraints (Simple Resource Network).
  - Uncertainty (pSTN, pSRN).
A solution is a candidate plan that:

- Satisfies all goal states and constraints in the QSP.
- Meets all specifications in the planning domain.
- Is executable.
For Over-subscribed Problems

• A planning problem maybe over-subscribed.
  – that is, no feasible plan can be found for it.

• Uhura can compute a set of **relaxations** to the problem that resolves the over-subscription.

• Relaxations apply to both goals and planning domains.
Types of Relaxations

✓ **Time**
  
  ‘Extend the assembly time by one hour.’

– **Resource**
  
  ‘Reduce the power consumption by 5 kWh.’

✓ **Risk**
  
  ‘Increase the acceptable risk level from 5% to 20%.’

– **Goal**
  
  ‘Assemble only one instead of two wingboxes today.’

– **Action**
  
  ‘Ask an off-duty staff to rivet the parts.’
Example Relaxations

- ‘Asking the technician to call in an off-duty staff’.

- ‘Asking the technician to rivet the wingbox’.
Approach: Research Questions

1. Collaborative plan diagnosis and repair.

2. Risk assessment and management.

3. Generate succinct explanations.
1. Collaborative Plan Diagnosis and Repair

• Over-subscription is caused by incompleteness and/or inconsistency.

• Diagnosis:
  – Incompleteness: find the subset of goals that are not supported by the planning domain.
  – Inconsistency: detect sets of goals and constraints that are in conflicts.

• Repair:
  – Find a set of relaxations that addresses all issues.
  – Restore the feasibility of the problem.
Possible Collaborations with Humans

• For approval to a candidate relaxation.
  – “Extend the assembly time by one hour, is it ok?”

• For preferences and constraints.
  – “How long does it take to repair the riveting robot?”

• For assistance.
  – “Can you rivet the wingbox before the staff arrives?”
Finding Good Relaxations

• Good relaxations are similar alternatives.

• Temporal and resource relaxations: similarity is measured by the linear distance.
  – “30-minute delay” is better than “3-hour delay”.

• Goal and action relaxations: similarity is measured by semantic relations\textsuperscript{[7]}.
  – “Chinese $\rightarrow$ Thai restaurant” is better than “Burger King”.
  – “Trader Joe’s $\rightarrow$ Star Market”.
  – “roast turkey $\rightarrow$ roast chicken”.
Key Elements of Uhura Architecture

CONVERSATIONAL & GRAPHICAL

USER INTERFACE

PLANNING PROBLEM

FEASIBLE PLAN

GENERATIVE PLANNER

CANDIDATE PLAN

CONSISTENCY TESTER

PREFERRED RELAXATION

INCOMPLETE

ADJUSTMENT

ADJUSTMENT

CONSISTENCY

ADJUSTMENT
Key Elements of Uhura Architecture

• **Planner:**
  – Generating complete plans.
  – Detecting *incompleteness*.

• **Tester:**
  – Evaluating plan activities, temporal and resource constraints.
  – Detecting *inconsistency* between goals and constraints.

• **Relaxation generator:**
  – Generating relaxations to resolve any incompleteness and inconsistency.
2. Risk Assessment and Management

• Pin-point the source of risk in a plan:
  – ‘Off-duty staff will arrive in 2 hours with only 80% chance.’

• Suggest trade-offs between risk-taken and performance:
  – “Accept 5% more risk of not having the staff helping assembly, so that I can complete the task on time”
Chance-constrained Temporal and Resource Problems

• Uncertainty exists in both time and resource.
  – “Bus is likely to arrive at 6:08, with a SD=2 minutes.”

• Chance-constrained probabilistic STN and SRN:
  – Uncertainty in both temporal duration and resource consumption/generation.
  – A risk-bound on the chance of violating any constraint.

< 5% Risk
Consistency Tester

- Evaluate the consistency of candidate plans:
  - If inconsistent, return a set of temporal and resource constraints that are in conflict.
  - The conflict also includes the chance constraint.
3. Generate Succinct Explanations

- **Uhura communicates:**
  - *explanations* on failures and resolutions.
  - *reasons* for the robot’s decisions and actions.

- Present only key ideas that users can draw conclusion from:
  - Include everything that is hard to find.
  - Omit less important or easy to infer results.

> Rivet the wingbox before the off-duty staff arrives.
Challenges in Generating Succinct Explanations

• Definition of ‘succinct’ explanation.

• Identify key ideas in the explanations for an over-subscribed problem.

• Identify what the users already know, and only present what they do not know.
Progress

Personal Transportation System

Flexible Manufacturing Test-bed

Commuter Advisory System

Deep-sea Expedition Plan Advisor

Application

Research

Resolving Over-constrained Temporal Problems

Resolving Temporal Problems with Uncertainty

Scheduling Chance-constrained Probabilistic Temporal Problems

Relaxing Chance-constraints to resolve Probabilistic Temporal Problems

Yu, Williams
IJCAI 2013

Yu, Fang,
Williams ICAPS 2014

Fang, Yu,
Williams
AAAI 2014

Yu, Fang, Williams
AAAI 2015
(submitted)
Research Questions

1. Collaborative plan diagnosis and repair.
   ✓ Temporal problems.
   – **Planning** problems

2. Risk assessment and management.
   ✓ Temporal uncertainty and risk.
   – **Resource** uncertainty and risk.

3. Generate succinct explanations.
Thesis Plan

• **Current time: 09/2014.**
    - Relaxing resource constraints: 01/2015 (IJCAI15).
Appendix

• Additional examples and technical details.
• Reference publications.
• More demos.
The objective function specifies the users’ preferences over the relaxations for the problem, and is defined over alternatives to the goals, constraints and actions.

Example: in the assembly example, different relaxations have different costs for the technician:
- Delaying the completion time costs $10,000.
- Calling in an off-duty staff costs $300.
- Joining the riveting himself costs $0.

This helps us prioritize the resolutions and simplify the interaction.
Applications

- **Peer-to-peer**: Human-Robot Collaboration in manufacturing tasks.

- **Supervision**: Trip advisor for WHOI deep-sea explorations, PTS and city commuters.
Remaining Work

• Plan generation and incompleteness detection.

• Resource uncertainty.

• Semantic Relaxation.

• Simple explanation.
• For each action, the duration and resource consumption may be random cannot be determined by Uhura.

• Example: transit from MIT to Logan airport may take any time between 10 to 100 minutes.

• We model such uncertainty using two approaches:
  – A simple set-bounded representation.
  – A probabilistic distribution representation.
Uhura enumerates relaxations in best-first order:
- It searches over subsets of constraints by making different variable assignments.
- It resolves a conflict by relaxing a constraint, partially and completely.
• Learning conflicts from controllability checking algorithms is more difficult.

  – For consistency checking, there is a **one-to-one mapping** between the distance edges and the bounds of constraints.

  – No such mapping exists for controllability checking (strong and dynamic) **due to the reduction procedures**, making it difficult to extract conflicts from the reduced graph.

• Key: during the reduction, **record the ‘contribution’** of each constraint and duration in the temporal problem.
A Strong Controllability Example

1. S1 → E1: A: [5, 10]
   - B: ≥ 4
   - C: ≥ 2

2. S1 ← E1: 10: A_U
   - -5: A_L
   - -4: B_L
   - -2: C_L

3. S1 ← S2: 10: A_U
   - -5: A_L
   - -4: B_L
   - 3: C_L, A_L
Resolving Uncontrollable Conflicts

- Constraint for resolving continuous conflict (negative value -1):

\[ \Delta C_L + \Delta B_L + \Delta A_L \geq 1 \]

where:

- \( \Delta B_L, \Delta C_L \) are relaxations for B and C.
- \( \Delta A_L \) is tightening for A.

and

\[ \Delta A_L \leq 5 \]
• Record supporting constraints for both \textit{requirement} and \textit{conditional} edges while generating the directed graphs.

\[
\begin{align*}
A - (x) & \rightarrow A' & \text{AB}_{\text{Lower}} \\
A' - (-x) & \rightarrow A & -\text{AB}_{\text{Lower}} \\
A' - (y - x) & \rightarrow B & -\text{AB}_{\text{Upper}} \\
A' - (b: 0) & \rightarrow B & -\text{AB}_{\text{Lower}} \\
B - (0) & \rightarrow A' & \text{None} \\
B - (B: x - y) & \rightarrow A' & \text{AB}_{\text{Upper}}
\end{align*}
\]

Learning Dynamically Uncontrollable Conflict

- **Record supporting** constraints and durations during the iterative reduction procedure.

- **Note that a constraint may be recorded multiple times during reduction.**

---

Upper-Case:  

- **A** \( \rightarrow \) \( B : x \) \( \rightarrow \) \( C \) \( \rightarrow \) \( y \) \( \rightarrow \) \( D \) \( \rightarrow \) \( \text{adds} \) \( A \) \( \rightarrow \) \( B : (x+y) \) \( \rightarrow \) \( D \)

- \( S_{CA} \) \( \rightarrow \) \( S_{DC} \)

Lower-Case:  

- **A** \( \rightarrow \) \( x \) \( \rightarrow \) \( C \) \( \rightarrow \) \( c : y \) \( \rightarrow \) \( D \) \( \rightarrow \) \( \text{adds} \) \( A \) \( \rightarrow \) \( x+y \) \( \rightarrow \) \( D \)

- \( S_{CA} \) \( \rightarrow \) \( S_{DC} \)

Cross-Case:  

- **A** \( \rightarrow \) \( B : x \) \( \rightarrow \) \( C \) \( \rightarrow \) \( c : y \) \( \rightarrow \) \( D \) \( \rightarrow \) \( \text{adds} \) \( A \) \( \rightarrow \) \( B : (x+y) \) \( \rightarrow \) \( D \)

- \( S_{CA} \) \( \rightarrow \) \( S_{DC} \)

No-Case:  

- **A** \( \rightarrow \) \( x \) \( \rightarrow \) \( C \) \( \rightarrow \) \( y \) \( \rightarrow \) \( D \) \( \rightarrow \) \( \text{adds} \) \( A \) \( \rightarrow \) \( x+y \) \( \rightarrow \) \( D \)

- \( S_{CA} \) \( \rightarrow \) \( S_{DC} \)

Label-Removal:  

- **A** \( \rightarrow \) \( B : x \) \( \rightarrow \) \( C \) \( \rightarrow \) \( \text{adds} \) \( A \) \( \rightarrow \) \( x \) \( \rightarrow \) \( C \)

- \( S_{CA} \)
Another Way to Resolve DC Conflicts

- A STNU is dynamically controllable if and only if it does not have a **semi-reducible** negative loop [Morris 2006].
  - We can resolve a conflict by **disabling** reductions that lead to edges in the negative loop.
Another Way to Resolve DC Conflicts

- We can resolve a conflict by **disabling** reductions that lead to edges in the negative loop*.

\[ \text{Relax the lowerbound of } A \text{ from 1 to 0 to disable the reduction.} \]

*A STNU is dynamically controllable if and only if it does not have a **semi-reducible** negative loop [Morris 2006].
Chance-constrained Relaxation

• No existing temporal feasibility checking algorithm applies to probabilistic problems.

• Key: ground the problem to a deterministic one with set-bounded uncertainty.
  – Then iteratively adjust the bounds to satisfy both chance and temporal constraints.
  – During the process, we can relax constraints to achieve
Example

- **Home → B**: Shop at B $\geq 35$
- **B → Y**: Drive B → Y
- **Y → Home**: Lunch at Y $\geq 75$
- **[0,180]**: Arrive home
Example

Resolving Over-constrained Temporal Problems with Uncertain Durations

- **Shop at B**: $\geq 35$
- **Arrive home**: $[0,180]
- **Lunch at Y**: $\geq 75$
- **Drive B → Y**
- **Y → Home**

**Home → B**
Example

Resolving Over-constrained Temporal Problems with Uncertain Durations
Conflict-directed Risk Allocation + Relaxation

[Diagram of a workflow with boxes and arrows]

Resolving Over-constrained Temporal Problems with Uncertain Durations
Conflict-directed Risk Allocation + Relaxation

Extract Conflict

\[ \text{Conflict}_1 > n_1; \]
\[ \text{Conflict}_2 > n_2; \]
\[ \text{Conflict}_3 > n_3; \]

... ...

Re-allocate Risk + Relax Constraints

\[ TC: \text{LB} \downarrow \text{ or } \text{UB} \uparrow; \]
\[ CC: \text{Risk bound} \uparrow. \]
• We start with a very conservative allocation, then iteratively tighten them to reach an agreement between temporal and chance constraints.
Formulation of the Optimization Problem

- We construct a non-linear optimization problem from known conflicts (linear constraints), chance constraint (non-linear constraints) and user preferences (linear objective).

**Temporal conflicts:**

\[
\Delta_{ShopB} + \Delta_{LunchY} + \Delta_{DriveH} \geq 30; \\
\Delta_{DriveB} + \Delta_{DriveBY} + \Delta_{Time} \geq 10; \\
\ldots \ldots
\]

**Chance constraints:**

\[
Risk(DriveB) + Risk(DriveBY) + Risk(DriveH) \leq CC; \\
\text{where} \ Risk(\alpha) = 1 - \int_{LB}^{UB} pdf(Duration(\alpha))
\]

**Objective function:**

\[
\text{minimize} \left( f(\Delta_{ShopB}) + f(\Delta_{LunchY}) + f(\Delta_{Time}) + f(CC) \right).
\]
• Uhura as a task scheduling assistant for WHOI scientists.

• http://youtu.be/yuVEUvFZENQ
• PTS with integrated Kirk, Uhura and pSulu for robust trip planning and execution.

• http://youtu.be/cxbYCrdd5ho4
Intelligent Manufacturing Testbed

• Kirk + Uhura + Pike for flexible assembly tasks.

• http://youtu.be/kjaEjvJnbXY
TATA Smart Grid Project

- (New) Household activity advisor for smart grid users.
  - A centralized controller determines how much power to supply to each household, given limited power supply and uncertainty.
  - Uhura communicates with residents of each household and manages their daily activities to meet the demand constraints while maximizing their convenience.
• A scientist (ST) is planning to deploy an under-water robot to survey a volcano eruption on the sea floor.
• After evaluating all the requirements, the Uhura (DA) determines that no solution meets all requirements.
• DA: I cannot meet all requirements due to the limited mission time and the uncertainty in eruption. Can you extend the mission to 4 hours and 10 minutes.
• ST: You can have at most 4 hours for this mission.
Risk Management in A Deep-sea Mission

- DA: May I increase the risk bound for this mission to 7.3% in order to meet the duration requirement?
- ST: I do not want to take that much risk on this task.
DA: Ok, can you shorten the traversal time from the site to the ship by 6 minutes? My plan can then cover 95% of the possible eruption time, between 8:45 and 10:51.

ST: That’s fine. Thanks.
• These questions are the keys for making Uhura an effective decision aid on collaborative plan diagnosis.
  – **Intelligent.**
  – **Reliable.**
  – **Conversational.**


References


