Integrating Acting, Planning, and Learning in Hierarchical Operational Models

Sunandita Patra\(^1\), James Mason\(^1\), Amit Kumar\(^1\), Malik Ghallab\(^2\), Paolo Traverso\(^3\), Dana Nau\(^1\)

\(^1\)University of Maryland, USA, \(^2\)LAAS-CNRS, France, \(^3\)FBK-ICT, Italy.

**Planning**
- Prediction + search
- To reach a goal or accomplish a task

**Acting**
- Performing tasks and actions in the real world
- Adapt to context, react to events
- Dynamic, partially observable environment
- Wrong move can lead to failures and dead ends
- Needs online help from planner

**Actor**

**Planner**

**Descriptive Models**
What the actions do?
- e.g., PDDL actions: action: action-identifier
  pre: test
  effects: effect, effect, ...
  effect

**Operational Models**
How to perform the task on actor's execution platform?
- method-name(arg\(_1\), ..., arg\(_n\))
  task: task-identifier
  pre: test
  body: computer program to generate commands and more tasks

**Planning Stage**

**Acting Stage**

**Problem:** The two models may not be consistent
- Can't verify or manage plans
- Acting suffers

**Our Contributions:**
- Planner UPOM that uses the actor's operational models for planning
- Learning strategies integrated with actor and planner

**Planning Algorithm: RAE**
RAE = Refinement Acting Engine
loop:
- for every new task
- Candidates ← (applicable method instances)
- choose m from Candidates
- create a refinement stack
- like a program execution stack
- initially with just task and m
- add the stack to Agenda for each task
- Progress(stack)

Use UPOM to make an informed choice

**Planning Procedure: UPOM**

**Learning Strategies:** Learn\(\pi\) and LearnH

**Learn\(\pi\):**
To choose a refinement method for a task

**LearnH:**
To estimate a heuristic for UPOM
- Gather training data from acting and planning traces of RAE and UPOM
- Train classifiers (multi-layered perceptrons)

**Experimental Evaluation**
Measured efficiency (reciprocal of cost) and success ratio in four simulated domains with different properties, such as, dead ends, concurrent tasks, dynamic events, sensing actions, agent collaboration, dynamic events.

**Conclusions**
- Using same model for both acting and planning is useful
- Key idea: Use operational models for planning instead of descriptive models
- Avoids inconsistency between actor and planner
- RAE with UPOM / Learn\(\pi\) / LearnH shows improved performance compared to purely reactive RAE in four simulated domains

**Contact:**
Sunandita Patra
patras@umd.edu
University of Maryland, College Park
https://sunanditapatra.wixsite.com/camp