Web-Planner
A Tool to Develop Classical Planning Domains and Visualize Heuristic State-Space Search

Maurício C. Magnaguagno, Ramon Fraga Pereira, Martin D. Móre, Felipe Meneguzzi
School of Computer Science (FACIN)
Pontifical Catholic University of Rio Grande do Sul (PUCRS)
Porto Alegre - RS, Brazil
{mauricio.magnaguagno, ramon.pereira, martin.more}@acad.pucrs.br, felipe.meneguzzi@pucrs.br
Introduction

- Classical planning
  - Declarative domain specification
  - Opaque intermediary steps
  - Challenging task for new users
  - Fixing mistakes is non-trivial

- Heuristic Functions
  - Modern classical planners
  - Different domains ⇒ different heuristic functions
  - Evaluate and select the best heuristic function
Introduction

● Planners have no easy setup
  ○ Academic projects
  ○ Small to no documentation

● No extra information
  ○ Planning failure gives no hint to the user
  ○ Is it impossible or incorrectly described?
  ○ How far the planner got until something went wrong?

● Solution
  ○ Move planner to the cloud (no setup)
  ○ Visualize internal data structures (explore)
# Background - Classical Planning

## Domain
- How the world “works”
- Predicates $\Rightarrow$ Features
- Actions $\Rightarrow$ Transitions
  - Preconditions
  - Effects
- Does the domain match the real world?

## Problem
- How the world is now
- Objects
- Initial state
- Goal state
- Is there a plan that reaches the goal?
Web Planner Architecture

- Interactions in the user-side
- Planning and data gathering in the server-side
- JSON as intermediate representation
Domain Development Interface
Visualization Interface - Search
Visualization Interface - Search

(1) Breadth First Search

(2) Best First Search with Hamming distance
Visualization Interface - Plan

- (clear peg2)
- (clear peg3)
- (clear d1)
- (on d3 peg1)
- (on d2 d3)
- (on d1 d2)
- (on d3 peg3)
- (on d1 peg1)
- (clear peg1)
- (on d1 peg2)
- (on d1 peg3)
- (clear d2)
- (clear d3)
- (on d1 d3)
- (on d2 peg1)
- (on d2 peg2)
- (on d2 peg3)
- (on d3 peg2)
Survey Results

The survey contained the following questions and answers (5 users):

How familiar are you with automated planning languages and algorithms?

● Have used PDDL before (1)

Did the visualizations help you to find any bugs/errors/interesting points during the course of your task?

● Found missing preconditions (1)
Survey Results

Mark other planners/tools you used in your experiments:

- Fast-Downard (1)
- JavaFF (1)
- JavaGP (3)
- Planning.domains (3)
- STRIPS-Fiddle (1)

Which features you missed the most?

- Support more requirements (2)
- Auto-complete (1)
- Option to clear console (1)
- Find (common) errors in PDDL (1)

System Reaction
Related Work

- Planning.domains
- myPDDL
- ...
Conclusions and Future Work

- Make planning easier to setup
  - PDDL editor with syntax highlight
  - Domain, problem and plan side-by-side
- Visible impact of heuristics
- Visible impact of actions

- Available at web-planner.herokuapp.com

- User-defined heuristics
- Selectable color schemes
- Side-by-side state-space view for comparison
- Better parsing messages
- Verify PDDL common mistakes
  - Missing/extra requirements
  - Missing free variables
  - Effect ⊆ Precondition
  - ...
- Larger user survey
- More planning instances available
- Define an API