Planning.Domains

Christian Muise
MIT CSAIL, USA
cjmuise@mit.edu

Abstract

Commonly used resources for the field of automated planning, such as benchmarks, problem generators, etc., are widespread over the internet. With `planning.domains`, we aim to (a) collect these resources in a central location; and (b) enable creative possibilities through a consistent interface to the larger planning community. In this demo, we focus on the three main pillars of `planning.domains`: (1) `api.planning.domains`– a programmatic interface to all existing planning problems; (2) `solver.planning.domains`– an open (and extendable) interface to planning-in-the-cloud; and (3) `editor.planning.domains`– a fully featured editor for planning domains.

Introduction

The inaugural International Planning Competition (IPC) was held in 1998.[1] Since that time, there have been 8 IPCs, each with their own set of benchmarks and problem compilers. While many of the contest websites are still available online, the benchmark problems used by the planning community are scattered and collected only in an ad-hoc manner for specific planners (for example on the websites for FD[2] and FF [3] planners).

The fundamental objective of the Planning.Domains (PD) initiative is to provide a set of resources, repositories, and tools for researchers to discover and develop planning problems. PD is made up of three principal components:

1. `http://api.planning.domains`: A programmatic interface to all existing planning problems
2. `http://solver.planning.domains`: An open and extendable interface to a planner-in-the-cloud service
3. `http://editor.planning.domains`: A fully featured editor for creating and modifying PDDL

The PD initiative was developed over the course of one year and announced at ICAPS-2015 in Jerusalem, Israel. It has since garnered the involvement of seven institutions, as well as financial support from the ICAPS organization. Ultimately, the aim for the PD initiative is to be a resource created both for and by planning researchers, as well as an avenue for outreach to other communities unfamiliar with planning technology.

[1] `http://ipc98.icaps-conference.org/`

[2] `http://hg.fast-downward.org/file/1b5bf09b6615/benchmarks`

[3] `https://fai.cs.uni-saarland.de/hoffmann/ff-domains.html`

In this demo, we will exhibit each of the components and their capabilities. For the remainder of the abstract, we will detail each of the components and briefly touch on immediate and long term future work for each of them.

api.planning.domains

The primary vision of planning.domains is to represent a canonical source for the existing planning benchmarks. However, there are a variety of issues that arise when one considers the naïve solution of a simple directory containing the existing benchmarks: what naming convention should be used? what is the hierarchy? how do you treat repeated domains over multiple years? and how do you partition the benchmarks? To address these and other concerns, we store the benchmarks in a version controlled publicly accessible repository, and *provide a programmatic access to the domains through an API*.

Direct database access is provided at three levels: (1) individual planning Problems; (2) a set of Problem objects for a single Domain; and (3) a set of Domain objects for a single Collection. Collections correspond to individual IPCs or commonly used benchmark suites. One advantage is the ability to define a canonical "all STRIPS IPC domains" collection that incorporates best practices such as using the most recent domains when there are duplicates, using corrected versions, sampling the domains so that roughly an equal number of problems exist for each domain, etc.

Domains represent a set of Problem instances, and naturally have their associated descriptions and origin information. Individual Problems have information about the correct domain and problem PDDL to use (which may be non-standard for some benchmarks), as well as statistics about the individual instances. These include best known lower bounds, upper bounds, classical width (Lipovetzky and Geffner 2012), etc. This enables queries such as "all problems where we do not know the optimal plan, but have width 1 and are thus trivial to satisfy suboptimally".

The API component of PD also comes with JavaScript and Python libraries to interface with the database, as well as a command-line utility to fetch and use the stored benchmarks. Moving forward, we have three key objectives for the API: (1) to expand the repository to alternative planning formalisms (FOND, POND, RDDL, RMPL); (2) to open the database to a curated form of statistics submission so that any researcher may contribute to the information on problems; and (3) to introduce a tagging mechanism for Problems, Domains, and Collections to allow for custom categories (e.g., identifying all delete relaxed problems, specify-

ing the requirements used in the modelling language, etc.).

solver.planning.domains

For many outside of the automated planning field, getting a planner to compile and run can be a daunting task. The Solver component of PD offers a planner-in-the-cloud service that can be invoked using a standard RESTful API – the PDDL is sent as raw text, and a planner running remotely returns a plan. The following Python code, for example, operates as an IPC-ready planner that accepts PDDL files and produces a plan using the service.[4]

```
import urllib2, json, sys

dom = open(sys.argv[1], 'r').read()
prob = open(sys.argv[2], 'r').read()
url='http://solver.planning.domains/solve'

data = {'domain': dom, 'problem': prob}
data = json.dumps(data)

req = urllib2.Request(url)
req.add_header('Content-Type',
               'application/json')
resp = urllib2.urlopen(req, data)
resp = json.loads(resp.read())

plan = []
for act in resp['result']['plan']:
    plan.append(act['name'])

with open(sys.argv[3], 'w') as f:
    f.write('\n'.join(plan))
```

Usage: ./planner.py domain.pddl problem.pddl plan.ipc

The planner is restricted to 10 seconds and 500Mb, but the project is open source and free for anyone to deploy on their own using different resource limits. The Solver initiative will also host a semi-regular contest to identify the most agile planner for use in the online service.

Future work for the project includes running the contest semi-annually, as well as improving the parsing capabilities for the remote planners: currently the ground action schema is returned, but further statistics may be of interest.

editor.planning.domains

The final PD initiative is an online editor for PDDL. The initiative is similar to previous efforts such as the model acquisition tool itSIMPLE (Vaquero et al. 2009), the PDDL Studio software (Plch et al. 2012), and the online basic editor myPDDL (Strobel 2015). The PD editor builds on the myPDDL editor in order to tie together the other PD initiatives, and to expand to include the features found in PDDL Studio and itSIMPLE. Among the standard features of an editor (e.g., syntax highlighting, bracket matching, and code folding), the following custom features have been developed:

[4]It is not recommended to submit this to an IPC, as it violates many of the rules for entered planners.

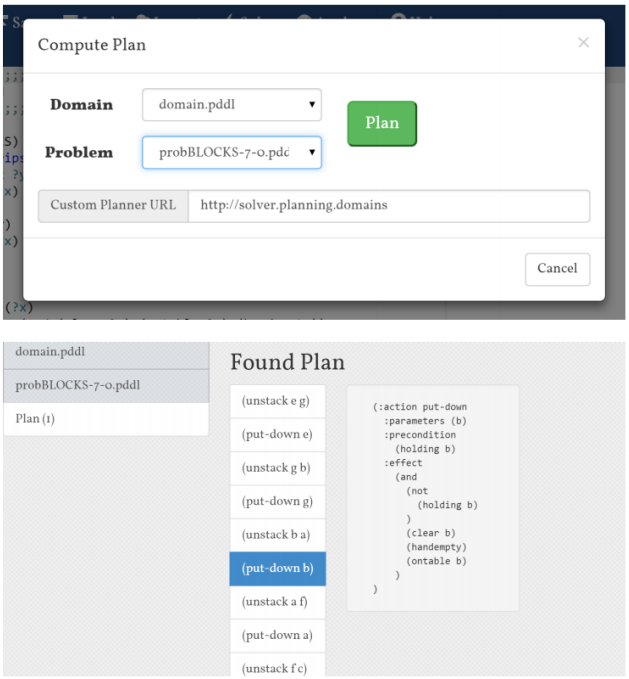

Figure 1: Online solver, and resulting plan

- The PD Solver is integrated so that solutions can be computed and displayed during editing (cf. Figure 1). Custom deployed solvers may also be used instead of the default.
- PDDL specific auto-completion can be used.
- Domain and problem files can be imported from remote servers by browsing through the PD API by Collection, Domain, and Problem.
- Problem analysis can be conducted using an online version of TorchLight (Hoffmann 2011); deployed using the same infrastructure as the open source PD Solver.

There are many features planned for the PD Editor, but most crucially the next step will be to introduce a plugin framework for others to implement features that expand the capability of the editor. Once the plugin framework is in place, development can be distributed, and ideas realized by one researcher can be released quickly to anyone using the PD Editor worldwide.

Summary

In this abstract, we summarized the three components that make up the PD initiative: (1) an API for existing benchmarks; (2) a remote planner-in-the-cloud; and (3) a fully featured PDDL editor. The purpose of the PD initiative is to provide planning researchers with a resource for existing and new benchmark domains, as well as to introduce researchers in other fields to the planning technology and formalisms that currently exist. In this demo, we present our progress on the former aspect. Moving forward, we aim to expand the capabilities of the PD initiative so that dissemination of planning research to other fields becomes commonplace.

References

Hoffmann, J. 2011. The TorchLight Tool: Analyzing Search Topology Without Running Any Search. In *Proceedings of the System Demonstrations, in the 21th International Conference on Automated Planning and Scheduling*, 37–41.

Lipovetzky, N., and Geffner, H. 2012. Width and Serialization of Classical Planning Problems. In *ECAI*, 540–545.

Plch, T.; Chomut, M.; Brom, C.; and Barták, R. 2012. Inspect, edit and debug PDDL documents: Simply and efficiently with PDDL studio. *System Demonstrations and Exhibits at ICAPS* 15–18.

Strobel, V. 2015. myPDDL - Knowledge Engineering for PDDL. `http://pold87.github.io/myPDDL/`. Accessed: 2016-03-18.

Vaquero, T. S.; Silva, J. R.; Ferreira, M.; Tonidandel, F.; and Beck, J. C. 2009. From Requirements and Analysis to PDDL in itSIMPLE3.0. *Proceedings of the Third International Competition on Knowledge Engineering for Planning and Scheduling, ICAPS 2009* 54–61.