Visual Planning Domain Design for PDDL using Blockly

Filip Dvorak, Anil Agarwal, Nikolay Baklanov
Schlumberger

Abstract

Industrialization of automated planning leads to a need to manage the life-cycle of planning models, and quite often the life cycle will include non-expert users, for whom the verbose logical models in planning languages are not the best communication instrument.
We are attempting to bridge the gap using popular drag-and-drop visual framework Blockly, developing an open-source tool that would seamlessly translate between Problem Domain Definition Language (PDDL) and Blockly.

Introduction

Industrial applications of Automated Planning necessarily lead to a need of a Planning Model Management System (PMMS) capturing the life-cycle of planning models, which typically includes the processes of:

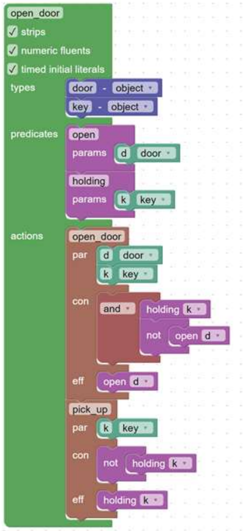
- motivation describing the real-world problem it captures,
- creation describing how the model has been created and how it can be re-created, be it a hand-made model, model generated from an ontology or machine-learned model,
- representation actual model itself using for example PDDL2.1 (Fox and Long 2003), and
- maintenance which includes continual testing of the model against several layers of automated tests, validation of the produced plans and version management controlling the performance of the model with new planners.

The PMMS is expected to be used not only by planning experts, but by a broad range of users, from subject-matter experts, who would be capturing their domain-expertise within a model, to non-technical product managers trying to communicate the content of a model to a broader audience.
We are following up on the initiative to develop tools that improve the learning curve and usability of automated planning technologies as a mainstream model-based decision making framework, such VSCode PDDL Extension (Dolejsi 2021). In particular, focusing on building a web-based opensource tool that would allow to develop and maintain planning models seamlessly transitioning between Blockly visualization and its ”drag and drop” design, and PDDL. Next sections elaborate on the two technologies.

Translation Example

```
(define (domain open_door)
(:requirements :strips :fluents :timed-initial-literals
:typing :negative-preconditions)
(:types door - object
                    key - object)
(:predicates (open   ?d - door)
                    (holding   ?k - key))

(:action open_door
  :parameters (?d - door ?k - key)
  :precondition (and (holding ?key)
      (not (open ?door)))
  :effect (open ?door))
  (:action pick_up
  :parameters (?k - key)
  :precondition (not (holding ?key))
  :effect (holding ?key))
)
```



Blockly

Blockly (Neil Fraser 2021) is a JavaScript open-source library developed by Google, which allows to define treenodes, visualize trees (directed graphs without cycles) and encode compatibility constraints between the tree nodes. Blockly has been growing in popularity, since tree-like languages are very common and the resulting framework is stable and mature, used across dozens of large projects. Blockly would struggle visualizing general graphs, e.g. trying to define a fully connected state-machine, so we only consider it for tree-like languages.

Planning Domain Definition Language

PDDL syntax has been originally inspired by the LISP syntax, as such, the projection between a Blockly tree and PDDL is straightforward to make and the main direction of development is to explore the range of most convenient ways how to define to Blockly blocks to maximize the efficiency of working with them. An example of an action defined in Blockly is shown in the Translation Example.

Conclusions

We are actively working on developing the Blockly-PDDL tool, first proof of concept can be found online (Dvorak 2021). In the future, we are planning to integrate the tool to VSCode PDDL Extension, extend the PDDL coverage to its later versions and also go beyond PDDL, supporting other action-modeling languages.

References

- Dolejsi, J. 2021. VSCode PDDL Extension. marketplace. visualstudio.com/items?itemName=jan-dolejsi.pddl.
- Dvorak, F. 2021. Blockly-PDDL One Day PoC. youtu.be/GoW51dO6SgU.
- Fox, M.; and Long, D. 2003. PDDL2.1: An extension to PDDL for expressing temporal planning domains. J. Artif. Intell. Res. (JAIR) 20: 61–124. doi:10.1613/jair.1129.
- Neil Fraser, Quynh Neutron, E. S. M. F. 2021. Blockly. developers.google.com/blockly
- Online Demo: **ai-planning.github.io/blockly-pddl/**
- Open-source repository: **github.com/AI-Planning/blockly-pddl**

Contact

Filip Dvorak
Schlumberger
Email: fdvorak@slb.com