

Historical developments in AI planning and search

Josip Matijević

April 2017

The goal of this document is to present a short summary of some of the important developments in the history of AI planning and search. The focus will be on describing the ideas that led to the development of Problem Domain Description Language.

STRIPS (Stanford Research Institute Problem Solver) was the first major planning system. It was developed by Richard Fikes and Nils Nilsson in 1971 [1]. An instance of a STRIPS problem is defined by the initial state of the world model, the available operators for modifying the world model, and the set of goals that need to be reached. For each operator we define its effects and preconditions which need to be fulfilled before the operator is applicable. A plan for a STRIPS instance is a sequence of operators that leads from the initial state to a state where all the goals are achieved.

ADL (Action Description Language) is an extension of STRIPS which enables encoding of more realistic problems. One example of a problem which cannot be modeled by STRIPS is a problem where the effects of actions depend on the situations in which these actions are performed. These kinds of STRIPS restrictions motivated the development of ADL. It was introduced by Edwin Pednault in 1987 [2] and has several differences compared to STRIPS [3]. ADL allows negative literals in states and has an Open World Assumption since the unmentioned literals are unknown, not false like with STRIPS. ADL allows conditional effects and the goals can involve conjunctions and disjunctions while STRIPS only allows conjunctions. ADL also has the equality predicate built in and supports types.

PDDL (Problem Domain Description Language) was introduced by McDermott and his colleagues in 1998 [4] as a standardized syntax for representing planning problems. It was mainly developed for the International Planning Competition (IPC) and has been constantly evolving with new extensions. IPC compares different planning systems on a set of benchmark problems which requires a standard language for specifying problems so PDDL provided a possibility to compare results from different planning systems. PDDL was inspired by STRIPS and ADL and contains sublanguages for them. A model of the planning system in PDDL contains two major parts: the domain and the problem definition. The domain definition consists of domain name, the definition of requirements, the definition of predicates and the definition of actions. Each action has parameters, preconditions and effects. The problem definition consists of problem name, the name of the connected domain, the definition of all possible objects, the definition of initial conditions and the definition of goals.

As already mentioned above, there are several extensions to the original definition of PDDL. At the moment of writing this document, the latest official version of PDDL is 3.1 which was first used at IPC in 2008.

References

- [1] Richard E. Fikes and Nils J. Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. In *Proceedings of the 2nd International Joint Conference on Artificial Intelligence*, IJCAI'71, pages 608–620, San Francisco, CA, USA, 1971. Morgan Kaufmann Publishers Inc.
- [2] E. P. D. Pednault. Formulating multiagent, dynamic-world problems in the classical planning framework. In M. P. Georgeff and A. L. Lansky, editors, *Reasoning about Actions and Plans*, pages 47–82. Kaufmann, Los Altos, CA, 1987.
- [3] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, December 2002.
- [4] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. PDDL - The Planning Domain Definition Language. Technical Report TR-98-003, Yale Center for Computational Vision and Control, 1998.