

# Planning Search Research Review

This short review consists of a historical report on different action and planning languages, and how they had an impact on the AI field as a whole.

*“Action languages are formal models of parts of the natural language that are used for talking about the effects of actions” (Gelfond and Lifschitz, 1998)*

## **STRIPS**

In 1971, Richard Fikes and Nils Nilsson from the Stanford Research Institute (SRI) created STRIPS (Stanford Research Institute Problem Solver), the first major planning system created and main planning component for SRI's Shakey robot project. The inputs for this planner were represented using a formal language named after the planning component. Given language represented planning problems through instances composed of an initial state, goal states and a set of actions. For STRIPS, states are a conjunction of positive literals, either propositional or first order literals. Conversely, actions are formed by a set of preconditions (which must be true for the action to be performed) and a set of postconditions (which describe how state is changed) (McCafferty, 2011).

STRIPS language was really attractive for researches of the era, which believed that it was a “on which to develop techniques that would be effective in more realistic models” (Fikes and Nilsson, 1993). It became very influential, to the point that it is the base for most of the languages representing automate problem instances nowadays.

## **ADL**

The Action Description Language (ADL) is another automated planning language, considered an improvement over STRIPS. Formulated in 1986 by Edwin Pednault, it extends the syntax of STRIPS action schemata, dropping some of STRIP's restrictions and allowing more complex problems to be encoded (Russell and Norving, 2010). It allows disjunctive, quantified, conjunctive and negative preconditions, besides conditional effects. Additional to increasing the domain encoding convenience, ADL reduces the size of domain descriptions needed (Coles and Smith, 2007). Having such extended language was advantageous during the time since it was possible to compile it to STRIPS, making it backwards compatible.

## **PDDL**

On an attempt to standardize planning domain and problem description languages, the Planning Domain Definition Language (PDDL) was created. Developed by Drew McDermott, it was mainly created to make the 1998 International Planning Competition possible, and has evolved since. Such an standardized language, PDDL has allowed a more direct comparison of systems and approaches, permitting a faster development on the field.

PDDL descended from several forebears, which is why it supports STRIPS, ADL and others, though not all planners fully implement PDDL. (Aeronautiques et. al., 1998). Many variations

and extensions of PDDL have been created since its origin having several extensions (Ida.liu.se, n.d.), including features like non-propositional state-variables (NDDL) and notation to create ontologies (MAPL).

## Bibliography

- Coles, A. and Smith, A. (2007). *Handling ADL*. [online] cs.cmu.edu. Available at: <http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume28/coles07a-html/node14.html> [Accessed 10 Mar. 2017].
- Fikes, R. and Nilsson, N. (1993). STRIPS, a retrospective. *Artificial Intelligence*, 59(1-2), pp.227-232.
- Gelfond, M. and Lifschitz, V. (1998). Action Languages. *Electronic Transactions on Artificial Intelligence*, 3, pp.195-210.
- Ida.liu.se. (n.d.). *Writing Planning Domains and Problems in PDDL*. [online] Available at: <http://www.ida.liu.se/~TDDC17/info/labs/planning/2004/writing.html> [Accessed 10 Mar. 2017].
- McCafferty, B. (2011). *STRIPS for Classical Plan Representation and Planning*. [online] Available at: <http://www.sharprobotica.com/2011/04/strips-for-classical-plan-representation-and-planning/> [Accessed 10 Mar. 2017].
- Russell, S. and Norving, P. (2010). *Artificial intelligence*. 1st ed. New Jersey: Pearson.