Historical Developments in AI Planning

In 1971, Fikes and Nilsson published a problem solver and formal language called STRIPS (**St**anford **R**esearch **I**nstitute **P**roblem **S**olver)[1]. This publication included a formal language called STRIPS which used first-order predicate calculus to describe a series of actions leading from an initial state to a goal state. STRIPS was designed to be used with the robot Shakey, which could solve problems in a physical environment. The STRIPS planning language represented the world as a set of conditions which could be true or false, and defined a set of operators (or actions) each of which had its own preconditions (conditions that must be true or false) and effects (making some conditions true and others false). Given an initial state and a set of operators, you could search the state space through possible actions to a given goal state.

Although STRIPS was very influential, especially the language it defined, shortcomings were found in subsequent research. In 1987 Edwin Pednault proposed a new language based on STRIPS but designed to remedy some of its shortcomings, which he called ADL (Action Description Langauge)[2]. ADL builds on STRIPS in the following ways: literals can be negated, the effects of actions can be conditional, testing equality between literals is a built-in language feature, and goals may include disjunctions (e.g., “cargo at Atlanta airport *or* cargo at San Francisco airport”). This increased flexibility allows the language to represent real-world problems that either couldn’t be represented in STRIPS, or would be impractical to represent. For example, the addition of conditional effects allows a single action to have different effects given the state in which it is performed[3].

Although ADL was found to be expressive enough to solve most classical planning problems, it lacked a standardized syntax used by everyone. In 1998, Drew McDermott and the AIPS-98 Planning Competition Committee needed a single planning language and syntax that could be used by all competitors to enable fair comparisons. Since no such standard existed at the time, McDermott and the committee created PDDL (Planning Domain Definition Language), which would be used in the competition they were planning[4]. This language was mainly based on ADL and another planning procedure UMCP[5].

These early developments in classical planning created the foundation which many other advancements in AI planning have built on. This is still an active area of research, and has a dedicated conference every year called ICAPS[6], which explores many applications of these techniques to real world problems.

---

[1] Fikes, Richard E. and Nilsson, Nils J. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, Technical Note 43R. AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, May 1971.

[2] Pednault. Formulating multi-agent dynamic-world problems in the classical planning framework. In Michael Georgeff and Amy Lansky, editors, Reasoning about actions and plans pages 47-82. Morgan Kaufmann, San Mateo, CA, 1987

[3] Edwin P. D. Pednault. ADL and the State-Transition Model of Action. Journal of Logic and Computation, Volume 4, Issue 5, 1 October 1994, Pages 467–512, https://doi.org/10.1093/logcom/4.5.467

[4] McDermott, Drew; Ghallab, Malik; Howe, Adele; Knoblock, Craig; Ram, Ashwin; Veloso, Manuela; Weld, Daniel; Wilkins, David (1998). "PDDL---The Planning Domain Definition Language". *Technical Report CVC TR98003/DCS TR1165*.

[5] Erol, Kutluhan & Hendler, James & S. Nau, Dana. (1994). UMCP: A Sound and Complete Procedure for Hierarchical Task-Network Planning. Proceedings of the International Conference on AI Planning Systems.

[6] http://www.icaps-conference.org