

# A History of AI Planning Systems

By Pankaj Mathur

AI Planning systems or Automated planning and scheduling systems is one of the major field of AI, which deals with dynamic unknown environments, unlike control and classification systems where planning can be done offline because environments are known to AI agents.

Historically, AI Planning systems were outcome of the research into state-space search programs, theorem proving methods, and control theory models. In addition to this, AI Planning systems were also heavily inspired by the practical needs of robotics domain, scheduling systems and other domains, seeking solutions for typical planning problem. To represent this planning problems from various domains, Researchers across the world, developed various planning modelling languages, in this article, we will explore few of popular AI planning systems and languages.

## Stanford Research Institute Problem Solver (STRIPS)

STRIPS is an automated planner developed by Richard Fikes and Nils Nilsson in 1971 at SRI International, it was first designed as the planning component of the software for the Shakey robot project at SRI (Stanford Research Institute).

STRIPS was modeled on previous state space search system called GPS (General Problem Solver) that used means-end analysis. The representational language used by STRIPS planner has much bigger impact on field of AI than its algorithms and is the base for the most of languages used to describe planning problems.

The *Classical Planning Language* used in STRIPS is composed of:

- **An initial state** is a conjunction of positive literals which cannot contain variables and invoke functions.
- **The goal state**, similarly to the state, is conjunction of positive and ground (no variables and no functions) literals.
- **A set of Actions** include pre-conditions and post-conditions. Both represented as a conjunction of function-free literals. Preconditions describe the state of world required to perform action, while post-conditions describe state of the world after action is executed

## Action description language (ADL)

ADL is an automated planning and scheduling system mostly used in field of robotics. It is considered an advancement of STRIPS. Edwin Pednault, an IBM Research Staff Member in the Data Abstraction Research Group, in 1987 proposed ADL system.

In ADL system, we assume that everything not occurring in the pre-conditions is unknown (Instead of being assumed false in STRIPS). ADL relaxed some of the STRIPS restrictions and make it possible to encode more realistic problems.

Also, ADL allows negative literals and disjunctions whereas, STRIPS allows only positive literals and conjunctions.

An ADL *schema* is composed of:

- **An action name**
- **An optional parameter list**
- **Four optional groups of clauses**
  - Precond
  - Add
  - Delete
  - Update

## Planning Domain Definition Language (PDDL)

PDDL is an attempt to standardize Artificial Intelligence (AI) planning languages. Originally PDDL was first developed by Drew McDermott and his colleagues in 1998. PDDL was also the official language for 1<sup>st</sup> and 2<sup>nd</sup> [International Planning Competition \(IPC\)](#) . A competition which was successful in establishing how well top-quality planning algorithms should actually perform.

PDDL was introduced as a computer-parsable, standardized syntax for representing planning problems and is heavily inspired by STRIPS and ADL systems.

According to PDDL manual (version 1.2), PDDL supports following syntactic features to support various, domain agnostics, AI Planning systems:

- Basic STRIP-style actions
- Conditional effects
- Universal quantification over dynamic universes (i.e. object creation and destruction)
- Domain axioms over stratified theories
- Specification of safety constraints
- Specification of hierarchical actions composed of sub actions and sub goals
- Management of multiple problems in multiple domains using different subsets of language features (to support sharing of domains across different planners that handles varying levels of expressiveness)

## References

- **Artificial Intelligence: A Modern Approach** by [Stuart Russell](#) and [Peter Norvig](#)  
<http://aima.cs.berkeley.edu/2nd-ed/newchap11.pdf>
- **PDDL – The Planning domain definition language version 1.2 Manual**  
<http://homepages.inf.ed.ac.uk/mfourman/tools/propplan/pddl.pdf>
- **Automated Planning and Scheduling**  
[https://en.wikipedia.org/wiki/Automated\\_planning\\_and\\_scheduling](https://en.wikipedia.org/wiki/Automated_planning_and_scheduling)
- **STRIPS** <https://en.wikipedia.org/wiki/STRIPS>
- **ADL** [https://en.wikipedia.org/wiki/Action\\_description\\_language](https://en.wikipedia.org/wiki/Action_description_language)
- **PDDL** [https://en.wikipedia.org/wiki/Planning\\_Domain\\_Definition\\_Language](https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language)