AI Planning was result of investigation into search state-space, Theron proving, control theory and other practical need like robotics. And in this review paper are going to go through some of the development in the field.

1. STRIPS

In 1971 Richard Fikes and Nils Nilsson at Stanford Research Institute developed an automated planner STRIPS (**S**tanford **R**esearch **I**nstitute **P**roblem **S**olver), the planner used formal language as input which was later referred to it as STRIPS. The language was Wilde adapted and became bases of other languages suck as ADL.

Where STRPS instance consist of
- An initial state.
- The specification of the goal states
- A set of actions, where each action include the following
  - Preconditions , which need to be satisfied before the action is performed.
  - Postconditions, which seed to be satisfied after the action is performed.

And a plan for such planning instance is a sequence of operator that can be executed from the initial state which will leas to the goal state.

Action Description Language (ADL)

In 1987 Edwin Pednault how is a specialist in the field of Data abstraction, proposed the language which considered advancement of STRIPS.

He extended the action definition of STRIPS actions by allowing quantified and conditional effects, and the precondition to allow for allow disjunctive, quantified and negative preconditions.

An ADL schema consists of

- An action name,
- An optional parameter list.
- Four optional groups of clauses labeled Precond,.Add, Delete and Update.

Where the Precond group is a list of formula that define the preconditions for the execution of an action. If the set is empty the value True is inserted into the group and the preconditions are always evaluated as holding conditions.

The Add and Delete conditions are specified by the Add and Delete groups, respectively. Each group consists of a set of clauses.

And here is an example

Action (
*Load (**c**: Freight, p: Airplane, A: Airport)*
*Precondition: At(**c**, A) ^ At(**p**, A)*
*Effect: ¬At(**c**, A) ^ In(**c**, p)*
)

Action (
*Unload (**c**: Freight, p: Airplane, A: Airport)*
*Precondition: In(**c**, p) ^ At(**p**, A)*
*Effect: At(**c**, A) ^ ¬In(**c**, p)*
)

Action (
*Fly (**p**: Airplane, from: Airport, to: Airport)*

*Precondition: At(**p**, from)*
*Effect: ¬At(**p**, from) ^ At(**p**, to)*
)

Planing Graph

In 1997 Avrim L. Blum and Merrick L. Furst published “Fast Planning Through Planning Graph Analysis” paper, which introduce new approach to planning in STRIPS-like domain based on contracting and analysing a compact structure they call it a Planning Graph.

The describe in the paper a new planner, “*Graphplan*", that uses this paradigm, which always returns a shortest possible partial-order plan, or states that no valid plan exists.

The idea is that rather then immediately start searching state-space like standard search, the algorithm instead start by explicitly constructing compact structure they call "*Planning graph*”, which encodes planning problem problem in such away that many useful constraint in the problem becomes available to reduce the amount of search needed. Unlike the state graph which can be huge, the Planning graph has polynomial size which can be constructed in polynomial time.

The “*Graphplan*” planner uses a planning graph to guide its search for a plan, the algorithm start with planing graph which only encodes the initial condition. And in state (i) takes the planning graph for state (i-1) and extends it one more time step, and then searches the extended planning graph for a valid plan of length (i), and the process continue until a solution is found.

References

[1] Artificial Intelligence A Modern Approach
[2] Wikipedia
[3] http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume28/coles07a-html/node14.html
[4] Fast Planning Through Planning Graph Analysis