Planning Domain Definition Language (PDDL)

The Planning Domain Definition Language was developed by Drew McDermott in 1998 as a means to made competing in the International Planning Competition possible. From then on there has been updates and revision to make the language more inclusive. The first iteration was inspired by STRIPS and borrows several elements. The standardization achievements of PDDL has made possible of having a common formal language with standard semantics. The first iteration described two major separations, domain and problem descriptors. The domain part describes the pieces or objects of the problem. It can also describe any heriarchy of the domain objects and predicates. Actions along with their precondictions and effects are also described within the domain. The problem descriptors described initial conditions and the defined goal states that the problem was trying to solve. Combining these two definitions, the overall model described the objects and capabilities of those objects as well as the initial and goal state orientations (or set) of them.

PDDL evolved every few years since inception introducing more features such as numeric fluents (a way to incorporate numerical descriptions along with objects and actions), derived predicates (such as assuming transitive relations between facts) and object fluents (object valued functions)

STRIPS

STRIPS or Standford Researh Institute Problem Solver was developed by Richard Fikes and Nils Nilsson in 1971. STRIPS is the base for most planning languages used currently. STRIPS is the incorporation of both an automated planner and language to feed the planner however the impact of STRIPS over time has been primarily the language to describe actions and states. A standard STRIPS implementation includes an initial states, list of goal state(s), set of possible actions and for each action a set of preconditions and postconditions. After a planner is successful in finding a plan, the plan is represented with a sequence of operators that can be executed in sequence from the initial state that leads to the goal state. A big benefit from using such a description language is that it is human interpretable.

Several planning languages use STRIPS as a base language to expand upon including PDDL as described above.

GraphPlan

Graphplan is an algorithm that consumes a problem definition expressed in STRIPS and trys to find a path of operations to reaching a goal state. The algorithm achieves this by maintaining facts as nodes and through a series of successive operations of expanding and searching, searches for goal state fluents. For each iteration of the algorithm, the graph will add another level to the graph by expanding the previous layer then immediately search for the goal state in the new layer and repeat. An introduction into graphplan however is the

maintenance of mutexes or exclusivity of potential nodes do to the fact of being incompatible of other present nodes. Incompatible facts and incompatible actions are deemed unqualified because of their paradoxical nature. This enables a pruning effect on nodes which increases performance of the algorithm.

References

https://en.wikipedia.org/wiki/Planning*Domain*Definition_Language https://en.wikipedia.org/wiki/STRIPS Artificial Intelligence, A Modern Approach, Russell, Norvig