AI Planning and Search: Historical Developments

3 major development in AI Planning and Search:

STRIPS “Stanford Research Institute Problem Solver” (Fikes and Nilsson, 1971)[1] was designed first as the planning component of the software for Shakey robot project at SRI.

High-level description:
STRIPS is representing a world model by a set of well formed formulas (wff) composed of:
- **states:** a set of offs describing the state of the world
- **goal state:** condition stated as a wff
- **a set of operators**, including a description of their effects and their preconditions wff schemata

Using STRIPS representation, we can construct a problem-solving tree by creating the sets of successors by applying applicable operators. Search strategy can then be employed to find a sequence of action from the initial state to the goal state.

Impact on AI Planning and Search:
The representation language used by STRIPS has been far more influential than its algorithmic approach[2]. It is now the basis for most of languages used in automated planning problem solvers.

PDDL “Planning Domain Definition Language” (Ghallab & al) [3] was introduced as a computer-parsable, standardised syntax for representing planning problems and has been used as the standard language for the International Planning Competition since 1998. It has been inspired by STRIPS and ADL (Action Description Language)

High-level description:
PDDL provided a formalism to describe:
- **the Domain**: with definitions of its relevant elements (requirements, object-type hierarchy, constant objects, possible actions, etc)
- **the Problem**: with definitions of its relevant elements (objects, initial conditions and goal states)

Impact on AI Planning and Search:
PDDL created a standard with clear semantics to be used for International Planning Competitions that enabled researchers to share and improve upon each iterations of their systems and approaches.
It is best summarised by the following quote “The adoption of a common formalism for describing planning domains fosters far greater reuse of research and allows more direct comparison of systems and approaches, and therefore supports faster progress in the field.”[4]

---

[1] Richard E. Fikes, Nils J. Nilsson (Winter 1971). "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving". Artificial Intelligence. 2 (3–4): 189–208. doi: 10.1016/0004-3702(71)90010-5.

[2] Artificial Intelligence: A Modern Approach by Norvig and Russell

[3] McDermott, Drew; Ghallab, Malik; Howe, Adele; Knoblock, Craig; Ram, Ashwin; Veloso, Manuela; Weld, Daniel; Wilkins, David (1998). "PDDL---The Planning Domain Definition Language

[4] Fox, M.; Long, D. (2002). "PDDL+: Modeling continuous time dependent effects". Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space

GRAPHPLAN[5] (Blum and Furst, 1995) is an algorithm using planning graphs to calculate reachability heuristics. It is building upon STRIPS and PDDL descriptions of a planning problems.

High-level description:
The planning graph is described by:
- alternating levels of nodes composed of states and actions, starting with a state
- 2 types of edges depending on the level:
  - from state to action: the state being the preconditions for the subsequent action
  - from action to state: the action's effect rendering a state True or False

Multiple heuristics can be derived from the planning graph:
For example:
- **Max-level**: maximum level cost of any of the goals
- **Level Sum**: sum of the first levels, at which any goal appears
- **Set-Level**: level at which all the literals in the goal appears without any pair being mutually exclusive

Impact on AI Planning and Search:
GRAPHPLAN[6] enabled the scale-up of modern planner performance by providing an extensible data structure to calculate reachability heuristics[7].

[5] A. Blum and M. Furst (1995). Fast planning through planning graph analysis. Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence

[6] Fox, M.; Long, D. (2002). "PDDL+: Modeling continuous time dependent effects". Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space

[7] D. Bryce and S. Kambhampati (2007) A Tutorial on Planning Graph–Based Reachability Heuristics. Association for the Advancement of Artificial Intelligence