Artificial Intelligence has come a long way. We often take granted what we consider to be AI because of how much with it now. Yet we should not forgot the advancements not only in what they can do, but how they can do so. The practices and methodologies that go into designing and AI program have come a long way in of itself.

The introduction of the **Planning Domain Definition Language** which was built upon STRIPS**,** developed by Malik Ghallab and his colleagues in 1998 served as a stepping stone in standardizing planning problems (Ghallab *et al*., 1998). The hope of PDDL was to encourage the open sourcing of AI in a format that everyone could understand and work from. Such a hope would help allow researchers to quickly pick up where other had left off, and accelerate the development of the field. It went as far as attempting to set a standards for notations at the AIPS-98 planning content (Ghallab *et al*., 1998).

In 1987, Richard E. Korf reported on the idea of **interleaving** of actions in his paper "Read-Time Heuristic Search: First Results", the idea that a complete planner should be broken down into smaller subgoals. The reason for this, Korf argued was due to the unrealistic computational time it took to completely solve a problem in a real world application. An example of such application would be a computer opponent in a chess game. Because there are exponential b^d possibilities to be calculated, as a player we cannot possibly wait for the computer to fully compute all the endgame moves before making it's first initial move (it would take several years!). Instead interleaving limits the scope of search to provide a short term goal so that it may make the best possibly decision given time constraints, allowing it to be more interactive in real-time.

Another prominent advancement in the field of AI was the **procedural net**. As Sacerdoti mentions in his paper "The Nonlinear Nature of Plans" (1975), plans can often be thought of in a linear manner, yet they are "not constrained by limitations of linearity" (Sacerdoti, 1975). Removing such assumptions and constraints from planning problems greatly reduce the complexity of a problem. Rather than representing a plan in a large long graph, we can represent a plan in a more compact graph where nodes (actions or states) can lead to another node with in the graph, just like the flow charts we can create in power points.

Each mile stone mentioned here has had a profound effect on the advancement of AI. PDDL helped researchers adhere to a standard, increasing the productivity of research and promote open sourcing algorithms. Interleaving allowed AI to become more interactive and applicable for real world problems. And procedural nets revealed to us the true nature of planning, that they nonlinear, helping reduce the complexity of solving planning problems.