

The Evolution of Planning Domain Definition Language (PDDL)

Inspired by STRIPS (Stanford Research Institute Problem Solver), ADL (Action description language) and others, was created with the intention of supporting the International Planning Competition [1].

With the time PDDL evolved generating multiple versions with the latest version being PDDL3.1.

MAPL

MAPL (Multi-Agent Planning Language), an extension of PDDL introduces non-propositional state-variables and actions than can be determined in runtime. MAPL also permits the evolution of agents for a common goal. As multi agent we denote any type of planning in multiagent environments, where in one side a planning process can be distributed by multiple agents, but also that individual plans can take into account concurrent actions from multiple agents [2].

OPT

OPT (Ontology with Polymorphic Types) based on the PDDL2.1 “is an attempt to create a general-purpose notation for creating *ontologies*, defined as formalized conceptual frameworks for domains about which programs are to reason” [3]. Being more elaborate type system, it allows users to make use of higher order constructs such as λ -expressions.

RDDL

RDDL (Relational Dynamic Influence Diagram Language) was created with intention to model classes that are difficult to model with PDDL[4]. Using RDDL is possible to model problems where there is concurrency, independent exogenous effects, continuing processes and non-goal rewards, distributions that are complex function of state or partial observability.

[1]. Wikipedia, https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language

[2]. Michael Brenner, *A Multiagent Planning Language*

[3]. Drew McDermott, *OPT Manual*, November 13, 2005

[4]. Scott Sanner, *Relational Dynamic Influence Diagram Language*