Research Review

Manasi Kulkarni

Automated Planning Scheduling is one of the most important field of AI along with Machine Learning, Natural Language Processing, Robotics, Computer Vision etc. Planning focuses on realization of strategies or action sequences. To accomplish given tasks, these systems need to have input data containing descriptions of **initial states** of the world, desired **goals** and **actions**. And the role of planning systems is to find sequences of actions which lead from initial state to given goal.

This review consists of a historical report on different action and planning languages, and how they had an impact on the AI field.

STRIPS

In AI, **STRIPS** (**St**anford **R**esearch **I**nstitute **P**roblem **S**olver) is an automated planner developed by Richard Fikes and Nils Nilsson in 1971 at SRI International. (Richard Fikes and Nilsson1974) This language is the base for most of the languages for expressing **automated planning** problem instances in use today; such languages are commonly known as **action languages**. It was the main planning component for SRI's Shakey robot project.

Given language represents planning problems through instances composed of an **initial state**, **goal states** and a **set of actions**. For STRIPS, states are a conjunction of positive literals, either propositional or first order literals. Conversely, actions are formed by a set of preconditions (which must be true for the action to be performed) and a set of postconditions (which describe how state is changed) (McCafferty, 2011).

STRIPS is a Classical Planning Language which was attractive for researches of the era, which believed that it was "on which to develop techniques that would be effective in more realistic models" (Fikes and Nilsson, 1993). It became so significant that it is the base for most of the languages representing automate problem instances nowadays.

ADL

Action description language (**ADL**) is an automated planning and scheduling system for robots. It is considered an advancement of STRIPS. Pednault, a specialist in the field of Data abstraction and modelling who has been an IBM Research Staff Member in the Data Abstraction Research Group since 1996 proposed this language in 1987. (IBM website) It is an example of an action language. ADL extends the syntax of STRIPS action schema, dropping some of STRIP's restrictions and allowing more complex problems to be encoded (Russell and Norvig, 2010). Unlike STRIPS, ADL doesn't assume that unmentioned literals are false, but rather unknown, what is better known as the Open World Assumption.

It allows disjunctive, quantified, conjunctive and negative preconditions, besides conditional effects. Additional to increasing the domain encoding convenience, ADL reduces the size of domain descriptions needed (Coles and Smith, 2007). Having such extended language was advantageous during the time since it was possible to compile it to STRIPS, making it backwards compatible.

PDDL

The **Planning Domain Definition Language** (**PDDL**) was first developed by Drew McDermott and his colleagues in 1998 (inspired by STRIPS and ADL among others) mainly to make the 1998/2000 International Planning Competition (IPC) possible and has evolved since. In other words, PDDL contains STRIPS, ADL and much more other representational languages. PDDL has allowed a more direct comparison of systems and approaches, permitting a faster development on the field.

Many variations and extensions of PDDL have been created since its origin having several extensions (Ida.liu.se, n.d.), including features like non-propositional state-variables (NDDL) create ontologies (MAPL), Ontology with Polymorphic Types (OPT), Probabilistic PDDL(PPDDL) etc.

Multi-Agent Planning

Multi-Agent Planning involves coordinating the resources and activities of multiple "agents".

According to NASA, multiagent planning is concerned with planning by (and for) multiple agents. It can involve agents planning for a common goal, an agent coordinating the plans (plan merging) or planning of others, or agents refining their own plans while negotiating over tasks or resources.

It involves how agents can do this in real time while executing plans (distributed continual planning). Multiagent scheduling differs from multiagent planning the same way planning and scheduling differ.

Some of the criterion important for MAP are computation costs, communication costs, plan quality, flexibility (commitment), robustness, scalability. Multi Agent Systems has many benefits especially in Robotics. It has applications in Industry (car assembly, workforce management etc.), Military (distributed sensors, unmanned vehicles etc.), Space (multiple rovers, spacecraft constellation, etc.), Games, Trading and so on.

(Referred from a NASA presentation https://ai.jpl.nasa.gov/public/home/bclement/papers/ma_planning_clement.pdf)

References:

Coles, A. and Smith, A. (2007). Handling ADL. [online] cs.cmu.edu. Available at: http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume28/coles07a-html/node14.html [Accessed 10 Mar. 2017].

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

Fikes, R. and Nilsson, N. (1993). STRIPS, a retrospective. Artificial Intelligence, 59(1-2), pp.227-232., 3, pp.195-210.

Ida.liu.se. (n.d.). Writing Planning Domains and Problems in PDDL . [online] Available at: http://www.ida.liu.se/~TDDC17/info/labs/planning/2004/writing.html [Accessed 10 Mar. 2017].

McCafferty, B. (2011). STRIPS for Classical Plan Representation and Planning. [online] Available at: http://www.sharprobotica.com/2011/04/strips-for-classical-plan-representation-and-plannin g/ [Accessed 10 Mar. 2017].

Russell, S. and Norvig, P. (2010). Artificial intelligence . 1st ed. New Jersey: Pearson.

A Nasa presentation
https://ai.jpl.nasa.gov/public/home/bclement/papers/ma_planning_clement.pdf

www, ibm.com