Project 3 – Research Review

During the late 1960s and early 1970s, an enthusiastic group of researchers (Richard **Fikes**, N.J. **Nilsson**, Cordell **Green** are among them) at the SRL AI Laboratory focused their energies on a single experimental project in which a mobile robot was being developed that could navigate and push objects around in a multi-room environment [1]. The cart was capable of rolling around an environment consisting of large boxes in rooms separated by walls and doorways; it could push the boxes from one place to another in its world. Its suite of programs consisted of those needed for visual scene analysis (recognize different objects), for planning (plan sequences of actions to achieve goals), and for converting its plans into intermediate-level and low-level actions in its world. In particular, it provided the context and motivation for development of the STRIPS.

Green's QA3[2] work focused our attention on the difficulties of describing in a formal logic the effects of an action, and particularly the difficulty of specifying those aspects of a situation that are not affected by an action (i.e. the "frame problem"). Those considerations produced what is arguably the key technical contribution of the STRIPS work, namely the STRIPS operator representation and the algorithm for modeling the effects of an operator based on the "STRIPS assumption" that a plan operator affects only those aspects of the world explicitly mentioned in the operator's deletions and additions lists.

Using **Fikes**'s GPS and REF-ARF problem solver[3] as out paradigmatic problem solving architecture, we needed to define meaningful "differences" between a situation described by a set of predicate calculus sentences and a goal situation in which a given predicate calculus sentence is true. Once differences were defined, we needed to specify what it meant for an operator to be "relevant" to "reducing" the difference. This technique was another key technical idea in the STRIPS design.

In retrospect, STRIPS was extremely limited in both the scope of planning issues it addressed and the complexity of problems it could solve. But the severe simplifying assumptions of the STRIPS framework had sufficient intuitive appeal to most researchers for them to believe that it was a viable foundation on which to develop techniques that would be effective in more realistic models.

The Action Description Language, or ADL[4] relaxed some of the restrictions in the STRIPS language and made it possible to encode more realistic problems. The Problem Domain Description Language of PDDL[5] was introduced as a computer-parsable, standardized syntax

---

[1] N.J. Nilsson, Shakey the Robot, SRI Tech. Note 323, Menlo Park, CA (1984).
[2] C.C.Green, Application of theorem proving to problem solving, in: Proceedings IJCAI-69, Washington, DC (1969) 219-239.
[3] R.E. Fikes, REF-ARF: a system for solving problems stated as procedures, Artif. Intell. (1970) 27-120
[4] Pednault, 1986
[5] Ghallab et al., 1998

for representing STRIPS, ADL, and other languages. PDDL has been used as the standard language for the planning competitions at the AIPS conference, beginning in 1998.

Planners in the early 1970s generally worked with totally ordered action sequences. Problem decomposition was achieved by computing a subplan for each subgoal and the stringing the subplans together in some order. This approach, called **Linear Planning**, was soon discovered to be incomplete. A complete planner must allow for **interleaving** of actions from different subplans within a single sequence. One solution to the interleaving problem was goal regression planning, a technique in which steps in a totally ordered plan are reordered so as to avoid conflict between subgoals. This was introduced by Waldinger (1975) and also used by Warren's (1974) **WARPLAN**.

Partial-order planning dominated the next 20 years of research, yet for much of that time, the field was not widely understood. The ideas underlying partial-order planning include the detection of conflicts (Tate, 1975a) and the protection of achieved conditions from interference (Sussman, 1975). The construction of partially ordered plans (then called task networks) was pioneered by the **NOAH** planner[6] and by Tate's **NONLIN** system[7]. An implementation of McAllester and Rosenblitt's algorithm called **SNLP**[8] was widely distributed and allowed many researchers to understand and experiment with partial-order planning for the first time.

Avrim Blum and Merrick Furst (1995, 1997) revitalized the field of planning with their **GRAPHPLAN** system, which was orders of magnitude faster than the partial-order planners of the time. A planning graph can be used in many different ways to guide the search for a solution.

Planning as satisfiability and the **SATPLAN** algorithm were proposed by Kautz and Selman (1992), who were inspired by the surprising success of greedy local search for satisfiability problems.

The resurgence of interest in state-space planning was pioneered by Drew McDermott's **UNPOP** program (1996), which was the first to suggest a distance heuristic based on a relaxed problem with delete lists ignored. The most successful state-space searcher to date is Hoffmann's (2000) **FASTFORWARD** or FF, winner of the AIPS 2000 planning competition. FF uses a simplified planning graph heuristic with a very fast search algorithm that combines forward and local search in a novel way.

---

[6] Sacerdoti, 1975, 1977
[7] Tate, 1975b, 1977
[8] Soderland and Weld, 1991