

# Evolution of Action Languages

Alex Staravoi tau

[alex.staravoi tau@gmail.com](mailto:alex.staravoi tau@gmail.com)

Arguably one of the most important historical developments in AI planning and search was a gradual evolution of action languages, allowing researchers and AI engineers to formally describe a problem as a state transition system and create a formal model of the effects of actions on the world. Three major milestones in converging on a universal action language could be represented by introduction of **STRIPS**, **ADL** and **PDDL** that were more or less built upon each other.

The first attempt to formalise and generalise planning problems was made by Richard Fikes and Nils Nilsson in 1971 when they introduced their **STRIPS** (Stanford Research Institute Problem Solver) automated problem-solving program [1, p. 189], which also included a language for formally describing a problem as a planner input. The way Richard Fikes and Nils Nilsson put it in their research paper, STRIPS planner searches a space of "world models" to find such a model where the specified goal is achieved. They expect every world model to have a set of applicable operators, where each of them transforms this world model into another world model, and the task of STRIPS problem solving framework is to find some composition of operators that transforms initial world model into the one satisfying goal condition [1, p. 190].

STRIPS problem-solving framework included both the program for finding solution as a composition of operators in the domain world and the language to describe the problem, and interestingly enough its formal description language became far

more influential than its algorithmic approach: most planning systems since then have used one variant or another of the STRIPS language [2, p. 409]. A good example of STRIPS advancement is Action description language, or **ADL**. It was proposed by Pednault, a specialist in the field of Data abstraction and modelling. ADL is also a planning and scheduling framework with its own action language, which is considered to be an advancement of STRIPS. Pednault's proposal consisted of three iterations of ADL, each one adding some relaxing properties to STRIPS: ADL-A, which main idea was allowing the effects of an operator to be conditional [3]; ADL-B, which introduced "static laws", a new kind of propositions allowing actions to be described with indirect effects; and finally ADL-C which also allowed propositions as static and dynamic laws with some more detailed characteristics [4].

The further advancement of STRIPS and ADL that turned out very influential and received many further successors, variants and extensions, was **PDDL** (Planning Domain Definition Language). It was introduced by Drew McDermott and his colleagues as a problem-specification language for the AIPS-98 planning competition [5], and has a distinct approach of separating description of the domain (or, the way they put it, *physics* of a domain) from description of a problem within that domain [5, p. 1]. PDDL has expressiveness of ADL for propositions, and syntactically supports basic STRIPS actions, conditional effects [5, p. 7] and more, which ensured PDDL to become a standardised syntax for representing STRIPS, ADL, and other languages [2, p. 409].

As those milestones depict evolutionary development of action languages building upon each other, evolution of formal syntax and language could seem insignificant in the wide history of AI development. But I think it should rather be treated as a foundation of all further advancements in the field, as ability to formally describe a domain world and a problem makes entire AI field easier to understand and approach, thus attracting intelligence that otherwise would not contribute to further field advancements.

## References

1. 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.
2. Russell, Stuart J.; Norvig, Peter (2003), *Artificial Intelligence: A Modern Approach (2nd ed.)*, Upper Saddle River, New Jersey: Prentice Hall.
3. Pednault (1987). *Formulating multi-agent dynamic-world problems in the classical planning framework*. In Michael Georgeff and Amy Lansky, editors, *Reasoning about actions and plans* pages 47-82. Morgan Kaufmann, San Mateo, CA, 1987.
4. Micheal Gelfond, Vladimir Lifschitz (1998) "*Action Languages Archived September 2, 2011, at the Wayback Machine.*", *Linköping Electronic Articles in Computer and Information Science*, vol 3, nr 16.
5. McDermott, Drew; Ghallab, Malik; Howe, Adele; Knoblock, Craig; Ram, Ashwin; Veloso, Manuela; Weld, Daniel; Wilkins, David (1998). "*PDDL--The Planning Domain Definition Language*" (PDF). Technical Report CVC TR98003/DCS TR1165. New Haven, CT: Yale Center for Computational Vision and Control.