Learning by Observation and Practice:  
A Framework for Automatic Acquisition of Planning Operators  
Xuemei Wang *  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh PA 15213-3891  
wxm@cs.cmu.edu  

The knowledge engineering bottleneck is a central problem in the field of Artificial Intelligence. This work addresses this problem in the context of planning systems. It automatically learns planning operators by observing expert agents and by subsequent knowledge refinement in a learning-by-doing paradigm. Our learning method is implemented on top of the PRODIGY architecture (Carbonell et al. 1992).

The learning system is given at the outset the description language for the domain, which includes the types of objects and the predicates that describe states and operators. The observations of an expert agent consist of: 1) the sequence of actions being executed, 2) the state in which each action is executed (pre-state), and 3) the state resulting from the execution of each action (post-state). Our planning system uses STRIPS-like operators and the goal of this work is to learn the preconditions and the effects of the operators. We assume that the operators have conjunctive preconditions and no conditional effects, everything in the state is observable, and there is no noise in the state.

The architecture for learning by observation and practice in planning includes the observation module, the learning module, the planning module, and the plan execution module, as illustrated in Figure 1.

Operators for the domain are learned from these observation sequences in an incremental fashion utilizing a conservative specific-to-general inductive generalization process. When an operator is observed for the first time, the system creates the corresponding operator such that its pre-condition is the complete pre-state and its effect is the difference between the post-state and pre-state. Operators thus learned may have extra preconditions that correspond to the irrelevant features in the state. These extra preconditions are removed incrementally if they are not present in the pre-states of the new observations. In order to further refine the new operators to make them correct and complete, and to evaluate the new operators, the system uses them to solve practice problems. The system first generates an approximate plan to solve the practice problem using the partially incorrect and incomplete operators, then it executes the plan. When an operator fails to apply, it repairs the plan and continues execution, until the problem is solved or a resource bound is exceeded. The system also refines operators based on the executions. In summary, operators are refined through a process which integrates learning, planning, plan repair, and execution. See (Wang 1994) for more details of this research. This learning method has been partially demonstrated in the extended-strips domain and the process planning domain. We are currently performing more extensive tests in these domains. We plan to extend the algorithm to handle situations when part of the state is not observable, and there is noise in the state.

References  