THE COORDINATION OF MULTIPLE GOAL SATISFACTION

Arthur M. Farley
University of Oregon
Eugene, Oregon
97403

In the well-developed problem solving system, much goal satisfaction activity will not involve the use of heuristic methods. Rather, a problem will activate a known, general strategy which, if successfully executed, guarantees goal satisfaction. To be applicable in a variety of contexts, such a strategy must be specified in a high level abstraction space. Solutions to tactical details, necessary to bind a strategy to an actual situation, will be based upon similar strategies at lower abstraction levels. The problem solving system will be expected to operate in situations where several, non-independent goals await satisfaction. To function effectively in such situations, the system must be able to coordinate its execution of the independently generated solution strategies.

There are two general approaches to strategy coordination. Both utilize recognized interactions between the current situation and states of the pending strategies. The first approach would combine the separate solution strategies into one, overall plan (Sacerdoti, 1975). The other approach would postpone coordination decisions, making appropriate action selections during execution of the strategies. This second approach has the advantage that new strategies can become pending at any time without replanning costs. Also, the system can take advantage of interactions between a current situation and states of the pending strategies which may not be foreseeable at the time or abstraction level of prior, overall planning. The aims of the research reported here are to define and solve problems associated with multiple goal satisfaction and to develop formalisms which facilitate effective execution coordination. What follows is a description of important aspects of an initial model of the action selection approach to strategy coordination.

The <u>current situation</u> is a set of propositions which describes the relevant, present state of the world. A (generalized) <u>strategy</u> is a tree of strategy states with goal state as root. A <u>strategy state</u> is made up of two component sets of propositions: SSN and SSO. SSN consists of propositions which no operator of the system can satisfy (i.e. weather conditions). Each proposition of the SSO component has at least one operator associated with it capable of adding a satisfying proposition to the current situation. A current situation which satisfies all propositions of a strategy state exemplifies that strategy state. Operators are defined in a usual way — as sets of add, delete, and precondition propositions. At higher levels of abstraction, the tactical specification and execution of an operator are problems to be solved using appropriate world models and other strategies.

A strategy is developed in a breadth-first manner from the goal state. A new strategy state may be linked to a given state by a relation, labelled with the name of an operator. The new state consists of the union of the preconditions of the operator with the given state, minus the propositions of that state's SSO component added by the operator. A proposed state with a non-empty SSO component is not added to a strategy if it is eguivalent to a prior state. A leaf state of the strategy tree has an SSO component which is empty or which can be produced only from prior states.

When a problem is presented, the strategy associated with the goal state is added to the set of pending strategies. The system coordinates execution of pending strategies by a cyclic process which initially classi'fies strategy states. A strategy state is classified as <u>relevant</u> if its SSN component is satisfied by the current situation. This reduces each strategy to a relevant substrategy. A relevant state is classified as <u>realized</u> if it is exemplified by the current situation. If all pending strategies are associated with independent goals, the system can select an operator leading from a realized state and then complete the cycle by specifying and executing that operator. Whenever a goal state is realized, its strategy is removed from the pending set.

However, the solution of a problem may require that a conjunctive set of goal states be satisfied by a current situation. Execution coordination of the associated strategies is then more complex. A state is classified as <u>critical</u> if it lies on all paths from leaf states to goal state of a pending, relevant substrategy. A strategy of a conjunctive set is classified as <u>ungrounded</u> if its goal state denies realization of a critical state of another strategy in the set. The goal state of that other strategy should be realized prior to the goal state of the ungrounded strategy. When a goal state of a conjunctive set is realized, strategies in the conjunctive set which are ungrounded solely due to interaction with the completed strategy become grounded. Grounded strategies not in the conjunctive set which have a critical state denying the newly realized goal state become ungrounded. When all goal states of the conjunctive set are real]'zed, such ungrounded strategies become grounded again. During each cycle, the system selects an operator leading from a realized state of a grounded strategy to a state not denying a realized goal state of a pending, conjunctive set. Criteria for operator selection include cost estimates and goal state priorities.

A program embodying the above model has been written to solve a class of block pile problems. Questions concerned with the satisfiability of conjunctive goal sets and with limited resources are under study.

<u>REFERENCES</u>

Sacerdoti, E. "The non-linear nature of plans" <u>Proc. IJCAI4,</u> 1975, p. 206-214.