GRASPER: A Permissive Planning Robot

Scott Bennett
SRA Corporation
2000 15th Street North
Arlington, VA 22201
bennett@sra.com (703) 558-7508

Gerald Dejong
University of Illinois/Urbana
405 North Mathews
Urbana, IL 61801
dejong@cs.uiuc.edu (217) 333-0491

Execution of classical plans in the real world can be problematic. Small discrepancies between a planner's internal representations and the real world are unavoidable. These can conspire to cause real-world failure even though the planner is sound and, therefore, "proves" that a sequence of actions achieves the desired goal. The difficulty, of course, is that the planner's proof is contingent on its internal world model precisely capturing all relevant features of the external real world. This is seldom the case, particularly in robotics where uncertainties abound. Small but unavoidable sensor errors preclude accurate knowledge of the state of the world Worse, the planner's own characterization of the effects of its actions are themselves only approximations. Real-world execution of a sequence of actions can introduce and quickly magnify inconsistencies with the internal micro world.

We have been investigating one response to this difficulty called permissive planning [Bennett, 1993; DeJong and Bennett, 1993; 1995], a machine learning extension to classical planning. Our video presents GRASPER, a permissive planning robotic system that learns to robustly pick up novel objects. In permissive planning, machine learning techniques are employed to refine the planning algorithm based upon empirical observations of success and failure in the world. Alteration of the planner is accomplished through planner bias adjustment. Planner bias is an inescapable facet of classical planning. It refers to the preference that a classical planner exhibits when it produces one particular plan from among the (often very large) set of distinct plans it could in principle construct. In the GRASPER system, planner bias is adjusted through Explanation-Based Learning of schemata which eclipse the native bias inherent in the conventional searching planner employed as the explanation engine.

The GRASPER system consists of a six degree of freedom Prab RTX scara-type manipulator, an overhead camera, a frame grabber, and an IBM RT computer running Lucid Common Lisp. GRASPER constructs plans to lift plastic pieces of a children's puzzle. No a priori models of the objects are given to the system. The pixel patterns from the camera are converted into simple polygons which serve as the object representations to the planner. These representations (like any internal representation of a real physical object) are flawed, being only approximations to the actual objects. GRASPER is given a conventional classical axiomatization for how objects can be surrounded, how closing the gripper applies a friction force between the fingers and the object, how sufficient friction establishes a grasp, how a grasped object can be manipulated, etc. The axiomatization, like the represented sensory data, captures the real world only approximately: The represented coefficient of friction is not precise. Operators that represent arm movement effects only approximate the motions they claim to perform. The forces that the robot fingers apply to a surrounded object also only approximate the operator's effects, and so on.

Using its initial knowledge, a plan is constructed to lift a designated object. Not surprisingly, the initial plan usually fails in the real world. Following the principle of permissive planning, the planning process, rather than the represented domain theory, is blamed for the shortcoming. The planning bias is adjusted. Through bias adjustment over several failures, the real-world effects of produced plans are made to conform to the projection of the original action sequence. The final pick-up schema can be interpreted as 1) squeezing harder than the world knowledge claims is necessary, 2) selecting grasp points along faces which are more nearly parallel, 3) selecting grasp points closer to the object's center of geometry than believed necessary, and 4) opening the gripper wider than believed necessary while approaching the target object.

A unique feature of permissive planning is that the planning algorithm rather than the underlying incorrect representations are adjusted to overcome execution failures. This is the dual of the more conventional approach which successively debugs the planner's domain knowledge, leaving the planning algorithm unchanged. As refined operator definitions and object representations become more elaborate, the complexity of the planning process can grow dramatically. One advantage of permissive planning is that the operator definitions and object representations remain as the implementor originally defined them.

Permissive planning is not conceived as a general solu-

*This work was conducted at the Beckman Institute at the University of Illinois at Urbana-Champaign. Funding was provided through the Office of Naval Research under grant N00014-91-J1563.

tion to all of the difficulties that arise in planning under uncertainty. However, it offers a unique set of advantages that can complement other approaches.

References

[Bennett, 1993] Scott Bennett. *Permissive Planning: A Machine-learning Approach to Planning tn Complex Real-world Domains.* PhD thesis, University of Illinois at Urbana-Champaign, 1993.

[DeJong and Bennett, 1993] Gerald DeJong and Scott Bennett. Permissive planning: A machine learning approach to linking internal and external worlds. In *Proceedings of the 11th National Conference on Artificial Intelligence,* 1993.

[DeJong and Bennett, 1995] Gerald DeJong and Scott Bennett. Extending classical planning to real-world execution with machine learning. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence,* 1995.