Decision-Theoretic Layered Robotic Control Architecture

Gilbert Peterson and Diane J. Cook

University of Texas at Arlington
Computer Science and Engineering
Planning and Learning Laboratory
Box 19015, Arlington, TX 76019-015
{gpeterson, cook}@cse.uta.edu

One of the current methods for developing task control software for robots is a layering approach. This approach generally consists of a symbolic planner, a task sequencer, and a behavioral robotic controller. The task sequencer is responsible for taking a command from an abstract plan and selecting which robot level actions and behaviors to execute. This representation leads to a robust functioning software control for a robot and a single task [Bonnasso and Kortenkamp, 1996]. When the robot must be reconfigured for a new task, elements must be added to the sequencer, and behaviors to the behavioral controller.

We are currently developing a decision-theoretic planner to function as the planning and sequencing layers for the architecture. It is our expectation that using a decision-theoretic planner as the sequencer will reduce the amount of work to reconfigure for a new task. We will verify the reconfigurability of our system by creating one set of behavior controllers for our robots and demonstrating the effectiveness of the controllers on multiple diverse plans. Nourbakhsh has implemented a similar system using a symbolic planner in which all planning was abstracted into three levels [Nourbakhsh, 1997]. We choose to incorporate a decision-theoretic planner instead of a symbolic one to make tradeoffs between risk and desire. In addition, this representation allows us to formally reason about the uncertainty that is inherent with robot tasks.

Our planner, DT-Graphplan, adds decision theory into the Graphplan algorithm, extending the domain to handle contingent and probabilistic actions as well as utility driven search. This is an extension of the recent work conducted on extending Graphplan to handle probabilities [Weld et al. 1998]. We incorporate utility reasoning into the existing multiple world approach that represents the effect of each action in all possible worlds. Instead of specifying a goal criterion, a minimum acceptable utility threshold is set for the planner. The planner searches for a plan that meets this minimum threshold, pruning world state with low utility values. Requests made to the robot are not represented as goals, but receive a utility commensurate with their value, and instigate replanning.

Certain elements in a robotic domain are best represented with decision-theoretic methods. One such element is resource manipulations associated with actions. For example, for a time-constrained task, the robot should perform actions to reach a goal, trading off taking great risks, or postponing a less important task to save time. The comparison between the risk of an action and its chance of success is another example. The utility of a risky move with low probability of success but great potential rewards will be compared with the utility of a conservative move with high probability of success and moderate potential rewards.

One issue that must be addressed regarding the decision-theoretic approach specific to robotic domains is associated with action looping. An example of the problem arises with sonar sensors. A percentage of error in sonar readings comes from specular reflections. If the robot remains in the same location and continues to read the sonar, this error may never resolve itself. However, each time the sonar is read, the belief in the validity of the reading increases to the point that an incorrect reading is accepted as valid. The sonar should instead be represented as two types of probability, an intermittent failure probability, and a non-looping failure. The non-looping failure prevents the probability of success from increasing each time the same action is immediately repeated.

Through decision-theoretic reasoning, our system can act as both the planner and sequencer in a robust, reconfigurable layered robot control architecture. The end goals are the creation of a decision-theoretic version of Graphplan, and a three layered robot control system where the planner handles each finer level of plan detail. Resulting in a system that can switch tasks with less programming, and plan in a resource dictated manner.

References

Bonasso, R.P. and Kortenkamp, D. 1996. Using a layered control architecture to alleviate planning with incomplete information. In Proceedings of the AAAI Spring Symposium, “Planning with Incomplete Information for Robot Problems”. 1-4.

Nourbakhsh, I. 1997. Interleaving Planning and Execution for Autonomous Robots. Ph.D. thesis. Department of Computer Science, Stanford University.

Weld, D., Anderson, C., and Smith, D. 1998. Extending Graphplan to Handle Uncertainty & Sensing Actions. AAAI-98. 897-904.