Integrating Uncertainty into Planners for Multi-Tier Autonomous Agent Architectures

J. P. Gunderson
University of Virginia, Dept. of Computer Science
Thornton Hall
Charlottesville, VA 22903
gunders@Virginia.EDU

Abstract

This paper addresses the issue of modifying planners to produce better plans in the face of uncertainty. The traditional planning paradigm has serious flaws in the face of a dynamic and uncertain world. As a result, integrated end–to–end mobile robotic systems must overcome numerous obstacles to integrate planners into a multi-tiered architecture. This paper explores the types of problems caused by uncertainty, and proposes several specific changes to be incorporated into planning systems. A proposal of future work is detailed.

Introduction

There seems to be a natural gulf between planning and acting. Many early systems in AI and robotics focused exclusively on the planning aspects, while other systems focused on reactive control systems. Extensive work has been done in each of these areas, yet end–to–end systems are still uncommon. In part this is due to inherent differences in the deliberative and reactive paradigms. A common approach to bridging these paradigms utilizes a multi-layered control architecture.

Conceptually, all that is needed is a mediator between the reactive control system and the deliberative planning of intentional changes to the world. This mediator layer is frequently designated as an *Executor* or *Scheduler*. Thus, the end–to–end system can be functionally decomposed into three modules: **Plan**, **Schedule**, and **Act**. The modular decomposition of the system often overlooks the highly coupled nature of the deliberative and reactive systems. This results in system degradation which stems from the partitioning into a reactive layer which deals well with the dynamic, uncertain nature of the environment, and a deliberative layer which is poorly designed to work with uncertainty. By modifying the representation used by the planner to encompass some representation of uncertainty, improved performance should be possible.

Background

The traditional reactive system(Brooks 1986) has as its basis a stimulus/response model. Sensors feed information about the world into the system, and this triggers specific behaviors, which utilize effectors to change the world. Rather than developing and maintaining a 'world-model', behaviors are enabled by having their conditions met by immediate sensor data. Higher level behaviors may be triggered to inhibit lower level behaviors. This allows the mobile agent to select which behaviors are appropriate, and to react effectively in a dynamic, uncertain world.

In contrast, the deliberative system attempts to maintain a sufficiently complete model of the world to allow planning and to produce sequences of actions to meet a goal(Fikes, Hart, & Nilsson 1972). In the classical versions, this abstract model of the world assumes both perfect knowledge and perfect operations: there is no uncertainty and operations are atomic and instantaneous. In this deliberative paradigm, a common metric of plan quality is "Does this plan change the world state from the initial conditions into the goal state, and does it do so in the minimal number of steps?"

With the introduction of uncertainty, the question becomes "How **likely** is this plan to achieve the goal state, given the uncertain nature of the world?". In an integrated end–to–end robotic system this metric is more appropriate. However, in many end–to–end systems the goal is to use an existing planner as–is and integrate it with an existing reactive system. Fusing a traditional planner and a reactive system into a single mobile agent architecture has been problematic(Gat 1992)(Hormann, Meier, & Schloen 1991)(Bonasso & Kortenkamp 1995).

This paper addresses modifying the planner to include notions of operator reliability and dynamic external events. Providing the deliberative layer with meta-data about the uncertainties of the real world can improve the planning process. Without this meta-data the deliberative layer can produce plans which have low probability of successful execution. Two problems which result are inefficient allocation of planner resources into unsuccessful plans, and the inability to choose alternative operations based on reliability. This paper presents some design considerations for an eventual end–to–end mobile robotic system, and outlines the intended development.

Aspects of Uncertainty

I will focus on six different aspects of uncertainty and their impact on the traditional planning approach. These are:

1. operators that fail by not achieving a desired state,
2. operators that fail by causing a needed condition to become false,
3. external events that achieve a desired state,
4. external events that cause a needed condition to become false,
5. sensor uncertainty that indicates a necessary condition which is currently true to be perceived as false, and
6. sensor uncertainty that indicates a necessary condition which is currently false to be perceived as true.

Each of these will be described, and examples presented, within the context of a specific world model.

In keeping with long established tradition, a suitable domain will be described. It will be designated "Pub-World", and will consist of a corner tavern-like environment in which the wait-staff consists of mobile robots (*bar-bots*). Since the proper functioning of this environment is complex and requires many precise operators to be applied, it provides numerous examples of operator failure. In addition, the often conflicting goals of staff and patrons provides for a significant variety of events external to the robotic system.

The next section discusses the issues of operator reliability, and external events from the viewpoint of integrating a deliberative system into a multi-layered architecture. The following section discusses specific changes that can be made to a traditional planner to incorporate knowledge about uncertainty. The final section presents some conclusions and discusses future work.

Planner Modifications to Improve Integration

The output of the deliberative layer is a plan that, when executed, alters the world to achieve some goal. The plan is a complete sequence of steps that must be executed. The sequence may be totally ordered or it may be partially ordered (a non-linear planner), however to achieve the current goal every step in the plan must be completed. This plan is based on two assumptions. First, when an operator is selected to achieve an action, it will work every time provided the explicit preconditions hold true. Second, if necessary preconditions hold during planning, they will also hold at execution time. Unfortunately, neither of these assumptions is valid for a mobile agent in an uncertain, dynamic environment. As a result plans fail.

Plan Failure

Plan failure can be defined as either reaching a state in a plan where the next operator in the sequence cannot be applied or completing the sequence without achieving the goal state. This can be due to either:

- some operation failing to change the world into state needed by a subsequent operator (operator failure), or
- some dynamic change to the state of the world invalidating a condition necessary to the goal state (external event).

The bar-bot rolls up to the wait-station to pick up a drink order. The gripper is positioned around the glass, and closes. The bar-bot lifts the gripper arm to place the glass on a tray. But the glass is wet, and the gripper slips off.

This is an example of an operator failing to achieve a necessary condition, which results in plan failure. At this point the system must either re-apply the operator, or report failure and wait for a new plan.

The bar-bot is placing six glasses onto the delivery tray. To accommodate all six, each glass must be placed precisely. However, due to slip (effector uncertainty) in the control arm, the fifth glass is placed incorrectly, making it impossible to place the sixth glass on the tray. The attempt to place the sixth glass fails.

In this case, due to sensor/effector uncertainty, the operator succeeded in its intended effect – placing the glass on the tray. However, it invalidated a necessary condition for the subsequent operator, and as a result the plan must be repaired.

Sensor uncertainty in this case can be considered as though it were an operator that failed by invalidating a required condition. In this case the action is reversible. The offending glass can be repositioned and the remainder of the plan can be executed. However, not all operator failures are reversible.

The bar-bot is placing the last glass of Guinness onto the delivery tray. As the stout foams over, the gripper slips. This causes the glass to drop to the floor and shatter. Since this was the end of the keg, the system cannot complete the drink order in a timely manner.

In this case the action is not reversible since the glass is destroyed. While there are other glasses, and presumably an additional keg, there will be considerable disruption to any existing plans, and, potentially, an unhappy patron.

External events can also impact the planning process in a dynamic environment. They can either impede plan or they can facilitate the plan completion.

The bar-bot needs to clear and wipe down a table after the patrons have left. The plan includes removing all the dirty glassware, removing the table setup, wiping down the surface, and resetting the table. While the gripper arm is positioning the wiping cloth, another patron walks by and places a *dirty glass on the table. This violates the precondition of a cleared table required by the wipe down operator.*

This external event results in the need to repeat steps in the plan, as though the corresponding operator had failed. However, since later steps have been executed, it is not appropriate to just 're-run' the plan, rather a new plan is needed.

The bar-bot needs to clear and wipe down a table after the patrons have left. The plan includes removing all the dirty glassware, removing the table setup, wiping down the surface, and resetting the table. However, the patrons have carried their glasses over to the dart board on their way out of the pub. As a result, there is no need to remove the glasses, and this step in the plan must be skipped.

Since the external event achieved the goal of removing the glassware from the table, the plan must be altered. In addition, one of the pre-conditions for the operator intended to remove the glasses was the presence of those glasses; this operator cannot be invoked. As a result the plan would fail.

The bar-bot needs to clear and wipe down a table after the patrons have left. The pub has added several new tables with glossy surfaces. After drying the table top, the bar-bot checks for excessive surface reflection - an indicator that the surface is still wet. The high gloss finish is interpreted as water on the table top, and the bar-bot dries the surface again and again. Finally, the system errors out, and the plan fails.

In this case, due to the dynamic nature of the environment and sensor uncertainty, a condition which was true (the dry table top) was perceived as false, causing the operator to fail. After repeated attempts, the plan failed, and a new plan was required.

Regardless of the cause, the deliberative layer must be able to deal with plan failure. There are two key issues which must be incorporated into deliberative layers: operator reliability and external events.

Operator Reliability

In the real world, attempts fail. If a deliberative system is to produce effective plans, it must have knowledge of the reliability of the operators it intends to apply. One approach suggests that after the plan is developed by the deliberative system, the execution system confirms the world state before the each step is executed. Only if the preconditions hold will the execution system implement the next step in the plan.

On the surface, the advantage of this approach is that no changes need to be made to the planner. The traditional planner is invoked with a starting world view, and it passes a completed plan to the execution system. If no operator failures occur, the plan executes normally. If one of the operators does fail, the execution system reinvokes the planner, using the new world view. It would seem that any planner can be integrated into an uncertain world.

Unfortunately, there is nothing to prevent the system from reaching an impasse, with the planner constantly suggesting an operator which just as constantly fails. Several systems address this issue by including *a priori* limits to the number of unsuccessful invocations of an operator(Haigh & Velosos 1998)(Gat 1997). Using reliability meta-data which is updated after each attempt, such an operator would have a very low reliability value. As a result, any plan which included it would have a low probability of success, and alternate, higher probability plans would be more likely to be developed. However, if this failure-prone operator were the only choice, some method similar to the maximum attempts might still be needed.

In cases where multiple operators can achieve the same goal current systems select a initial operator either using context-based rules(Schreckenghost *et al.* 1998) or an *a priori* selection process. However, the reliability meta-data would support the ability to select the operator most likely to succeed. This would result in plans with higher probability of overall success(Wilkens *et al.* 1995).

A second difficulty lies in the concatenation of operators. As the number of steps needed to complete a plan grows, the cumulative probability of the plan succeeding drops. In the AI Planning Systems 1998 Competition some sample problems required over 100 individual steps to complete a solution(McDermott 1998). If each step were independent and had a reliability of 99% such a plan would have a successful implementation rate of only 37%.

A traditional planner attempts to produce an optimal plan, where optimal is defined as minimal number of steps. However, since some operators will be more reliable than others, a longer plan might be far preferable to shorter, less reliable plans. In a planner with no concept of operator failure, there is no way to represent this. A breadth first search, or satisfiability algorithm might produce consistently shorter plans, but ones with lower probability of success. Since a planner with operator reliability information will be able to optimize the likelihood of plan success, the mobile robotic system will successfully complete more plans.

External Events

In the real world, things change. If a deliberative layer is to develop successful plans, these plans must be aware of the dynamic nature of the environment. Reactive systems are based on the concept that a condition it true only as long as it is being perceived. The certainty of a condition approaches 1.0 while it is being sensed, and gradually decays towards 0.0 as external events occur. Planners assume permanence as a default, which is unrealistic in an environment where sensors fail, sensor data analysis can be incorrect, and independent events can change the state of the world.

It would seem that the effects of the dynamic world can be managed by the Scheduler/Execution system and no changes need to be made to the planner. The reactive system could automatically sense to confirm the preconditions of an operator before invocation, and the scheduler could detect that the current state obviates the need for a particular operator(Drabble 1993). However, there are several problems with this. First, there is no guarantee that the preconditions are perceivable from the location, or at the time, at which the operator is applied. This is a direct result of the nature of the reactive system, since it does not store data about the state of the world. Second if external events cause a necessary condition to become true, the scheduler may not be capable of optimizing around operations which were intended to achieve that condition. Finally, given the dynamic world in which many mobile agents function, complex plans can be out of date by the time they are executed. Making precise long-range plans in a dynamic world requires sinking resources into high-risk, low-return investments. Planners could utilize those computational resources more effectively(Ingrand & Georgeff 1990). By attempting to use a traditional planner, end–to–end systems will frequently be planning for conditions which never occur.

Altering the Planner

The arguments above suggest that unless the planner has meta-data about the reliability of operators, the plans produced can be sub-optimal in terms of plan success, and can require more processing resources than necessary. By incorporating operator reliability information in the planner, planning can be improved, and plan success rates increased. Clearly, the overall system could benefit if this meta-data is incorporated into the deliberative system in a robotic application. This benefit will come at a cost in increased code size to store the meta-data, and increased processing overhead to utilize the meta-data. Design decisions must be made with both costs and potential benefits in mind.

Modeling Uncertainty

If planners have reliability meta-data, they can plan more effectively. Integrating uncertainty into a planner can be accomplished by adding a reliability data structure to each operator, which records the number of invocations of the operator, and the number of successful executions. Upon every invocation of the operator, the execution system returns a success/fail code. This is used to update the operator meta-data. If the planner supports compound operators (e.g. 'chunking' in SOAR(Laird *et al.* 1991), or MACOPS in Strips(Fikes, Hart, & Nilsson 1972)) these compound operators would have their own reliability meta-data, which can used to track the interactions between operations in a sequence.

As conditions in the environment change, the reliability of operators will also change, for example adding rubber pads to gripper surfaces would increase the reliability of the 'pick-up' operator. This dynamic updating can be kept current using either exponential decay filtering, or Kalman Filtering, to integrate previous reliability with more recent performance.

In this paper I propose using a single reliability value for each operator. However, it can be difficult to capture the reliability of a particular operator in a single value, since different situations can produce significant variation in the failure rates. A robot moving down a hallway might have a 99.99% reliability at 11 p.m. when the hallway is deserted, but just before lunch the reliability would be far lower. A mechanism of modeling temporally varying reliability has been implemented(Haigh 1998) by incorporating more complex reliability measures, which are temporally coded. While this supports situated reasoning(Georgeff 1991), it also would increase the complexity of the meta-data, and increase the overhead of determining reliability.

Planning for Failure

As a planning space is being explored, situations arise where more than one operator can be used to achieve a necessary condition.

> *A group of six patrons have ordered dinner. It is possible to place all six plates onto a large tray and deliver them to the table in one trip. An alternate plan uses a smaller tray that is easier to maneuver through the pub, but which will require two trips, and considerably more time.*

The inclusion of operator reliability meta-data provides the deliberative system with additional tools to select between operators. A plan can be assigned a probability of success during the planning process. As the plan is built out of a sequence of operators, a cumulative probability value can be calculated. A plan consisting on no operators would have a success value of 1.0. As operators are included in the plan the probability of successful completion would drop, corresponding to the likelihood of failure of each new step in the plan. During operator selection, the deliberative layer can estimate the likelihood of success of a proposed plan, and choose between the candidate operators to optimize the probability of plan success.

If the planning system generates partially ordered plans, the calculation is the same, provided there is no interaction between the operators. While the plan representation is non-sequential, all steps in the plan must be executed, and so the overall plan reliability can be calculated. If there is interaction between the operators, then the order does matter, and the problem becomes more complex. The system must be designed with the specific type of planner in mind to prevent an unsupportable increase in planning overhead(Boddy & Dean 1994).

When coupled with hierarchical planning(Durfee & Lesser 1988), the system can produce an abstract sequence of plan stages(Fikes, Hart, & Nilsson 1972), spanning from the current perceived world state to the desired goal state. These plan stages are abstract, and will need to be expanded into sequences of executable operators. However, the smaller number of plan stages reduces the size of the abstract plan to manageable levels(Sacerdoti 1974). If these abstract plan stages also support reliability meta-data, a complete plan can be selected to increase the probability of overall plan success.

As each stage is elaborated into more detailed steps, the estimate of plan success can adjusted for every operator in the plan. This detailed planning can be terminated when the estimated success of a sequence of operators drops below some threshold, and that portion of the abstract plan which is fully elaborated can be passed to the execution system. This frequently is augmented by the use of *asynchronous planning*(Wilkens *et al.* 1995) where planning continues on the deliberative system during the execution of a plan by the reactive system.

If an operator fails, planning resumes on the abstract level, until a new complete abstract plan is produced[1]. This interleaving of planning and execution allows the system to be more responsive on two fronts. First, the planning task is broken up into smaller parts, reducing the planning space and complexity. Second, given the dynamic, uncertain nature of the real world, the planner can adjust the plan to take advantage of external events which move towards the goal state, and plan around external changes which impede the plan.

The effectiveness of interleaved planning is controlled in part by the choice of the mechanism for establishing the abstract plan stages. If the deliberative system has knowledge about the reliability of the operators, this knowledge can be used to delimit abstract stages whenever the combined probability of successful plan completion drops below a threshold, or by any other appropriate mechanism. As a result an adaptive system is created, which can attempt to reduce the overhead and increase the reliability of the mobile agent.

Conclusions and Future Work

In this paper I have presented arguments that planners can benefit from a model of uncertainty. Mobile robots function in environments where all operators have a possibility of failure. With knowledge of the reliability of operators the deliberative system can produce plans which are more likely to succeed. In addition, the cumulative probability of plan success can be used as an adaptive method to interleave planning and execution.

For mobile robots in dynamic environments, deliberation consumes scarce computational resources. Including representations for both operator success and for the impact of external events allows the deliberative system to minimize planning resources committed to low probability events.

[1]This assumes that the failure leaves the system is a state from which recovery is possible.

I have suggested two simple changes to traditional planners which can be used to capture this uncertainty: meta-data about operator reliability and aggressive use of interleaved planning and execution. Reliability meta-data can be a simple as prior probability information and still provide significant benefits to the deliberative system. This meta-data consumes little space and can be maintained and utilized with low computational overhead. In addition, it can provide a framework for adaptive interleaving of planning and execution.

The next step is to incorporate these changes into an end–to–end mobile agent application.

There are four primary areas of future work:

- formalize the representation of operator reliability,
- model the mechanism for operator success/failure updates,
- implement these changes in an integrated mobile robot, and
- benchmark the resulting impact on the effectiveness of the end–to–end system.

The current representation of reliability uses the prior probability (or naive forecast) of operator success. This is an unconditional probability, and incorporating Bayesian posterior probabilities may provide improved planning capabilities. So one area of future work is investigating the potential benefits of a more sophisticated situated probability measurement.

If this system is to be adaptive to changes in the operator reliability, a more formal notion of the success or failure of an operator is required. In addition, it will be necessary to implement a mechanism to communicate this meta-data back to the deliberative system that does not put a burden on the reactive system, or consume excessive bandwidth in the agent. It will also be necessary to develop an updating mechanism which consumes minimal resources.

After completing these designs, a development of a complete end–to–end system will be undertaken using modified existing planning software and robotics platform.

Finally, comparisons will be made to benchmark probabilistic planners and equivalent planners without enhancements for uncertainty, to determine the costs and benefits of including uncertainty in the planning system.

References

Boddy, M., and Dean, T. L. 1994. Deliberation scheduling for problem solving in time-constrained environments. *Artificial Intelligence* 67:245–285.

Bonasso, R. P., and Kortenkamp, D. 1995. Characterizing an architecture for intelligent, reactive systems. In *AAAI-95 Spring Symposium.*

Brooks, R. A. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* RA-2(1):14 – 23.

Drabble, B. 1993. Excalibur: a program for planning and reasoning with processes. *Artificial Intelligence* 62(1):1–40.

Durfee, E. H., and Lesser, V. R. 1988. Incremental planning to control time–constrained blackboard–based problem solver. *IEEE Trans. on Aerospace and Electronic Systems* 24(5):647–662.

Fikes, R. E.; Hart, P. E.; and Nilsson, N. J. 1972. Learning and executing generalized robot plans. *Artificial Intelligence* 3(4):251–288.

Gat, E. 1992. Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile agents. In *AAAI-92*.

Gat, E. 1997. Esl: A language for supporting robust plan execution in embedded autonomous agents. www_aig.jpl.nasa.gov/public/home/gat/aero97.html.

Georgeff, M. P. 1991. Situated reasoning and rational behavior. Technical Report Technical Note 21, Australian Artificial Intelligence Institute.

Haigh, K. Z., and Velosos, M. M. 1998. Interleaving planning and robot execution for asynchronous user requests. *Autonomous Robots* 5(1):79–95.

Haigh, K. Z. 1998. *Situation–Dependent Learning for Interleaved Planning and Robot Execution*. Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University. CMU–CS–98–108.

Hormann, A.; Meier, W.; and Schloen, J. 1991. A control architecture for an advanced fault-tolerant robot system. *Robotics and Autonomous Systems 7* 211–225.

Ingrand, F. F., and Georgeff, M. P. 1990. Managing deliberation and reasoning in real-time ai systems. In *Proceedings of the 1990 DARPA Workshop on Innovative Approaches to Planning*.

Laird, J. E.; Yager, E. S.; Hucka, M.; and Tuck, C. M. 1991. Robo-Soar: An integration of external interaction, planning, and learning using Soar. *Robotics and Autonomous Systems* 8:113–129.

McDermott, D. 1998. Artificial intelligence planning systems – 1998 planning competition results. http://ftp.cs.yale.edu/pub/mcdermott/aipscomp-results.html. Accessed: July 21, 1998.

Sacerdoti, E. D. 1974. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence* 5:115–135.

Schreckenghost, D. L.; Edeen, M. A.; Bonasso, R. P.; and Erickson, J. 1998. Intelligent control of product gas transfer for air revitalization. In *Proc. 28th Intl. Conf. on Environmental Systems - 1998*.

Wilkens, D. E.; Myers, K. L.; Lowrance, J. D.; and Wesley, L. P. 1995. Planning and reacting in uncertain and dynamic environments. *Journal of Experimental and Theoretical Artificial Intelligence* 7(1):121–152.