Planning for Murphy's Law: Uncertainty in CIRCA

David J. Musliner & Kurt D. Krebsbach
Automated Reasoning Group
Honeywell Technology Center
3660 Technology Drive
Minneapolis, MN 55418
{musliner,krebsbac}@src.honeywell.com

Abstract

Anything bad that can happen will happen. Plan accordingly.

Introduction

Many planning and control systems attempt to represent their "degree of uncertainty" and incomplete information using quantitative measures (e.g., probabilities) or other techniques. Interestingly, our work has shown that this level of detail is relatively unimportant in mission-critical domains. When certain types of events and conditions are considered catastrophic (and hence wholly unacceptable), degrees of uncertainty become a moot point: Murphy's Law must be observed and planned for. Any *possible* way of reaching a catastrophic failure condition must be planned for and eliminated in order to provide guarantees of safe system performance.

The CIRCA architecture was designed to provide predictable real-time performance and guaranteed system safety in mission-critical domains. The current implementation of CIRCA embodies a wide variety of techniques for dealing with uncertainty and incomplete information in several forms. In particular, CIRCA builds plans that can tolerate:

- Uncertainty in the timing characteristics of actions and exogenous processes.
- Uncertainty in action consequences.
- Uncertainty in future goals.
- Uncertainty in system state.
- Uncertainty in initial conditions.

In addition, the CIRCA model of planning explicitly defines a notion of "completeness" for plans used in interleaved planning and execution. This definition justifies CIRCA's claims to real-time performance guarantees.

In the following sections we provide a brief overview of the CIRCA architecture followed by additional details on how each of these types of incomplete information is handled in the CIRCA approach.

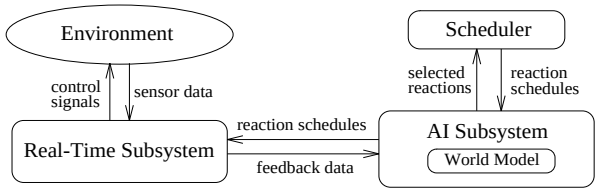

Figure 1: The Cooperative Intelligent Real-Time Control Architecture.

Overview of CIRCA

As illustrated in Figure 1, CIRCA consists of several parallel subsystems. The AI Subsystem (AIS) is responsible for using AI planning methods to reason about a world model, deriving appropriate monitoring and control reactions for the system. These reactions are built into an execution schedule by the Scheduler module, and then downloaded to the Real-Time Subsystem (RTS). The RTS is designed to provide a predictable execution environment which can enforce hard real-time response guarantees for the planned reactions. The RTS executes previously-derived plans while the AIS and Scheduler are cooperatively developing a new plan; each reaction plan is designed to keep the system safe (avoiding failures), so that the search-based planning performed by the AIS is isolated from the ongoing real-time deadlines of the environment.

The world model and planning algorithm that the AIS uses to develop reaction plans are detailed in [2]. For our purposes, it is sufficient to understand that the model is a modified state/transition graph in which states correspond to complete descriptions of the world, and three types of transitions represent the ways the world can change. *Temporal transitions* represent time and ongoing processes. The timing behavior of a temporal transition is related to the rate of the process it represents: for example, the process of moving through a doorway will take some minimum amount of time to complete, depending on the rate of travel. *Event transitions* represent occurrences outside the

Appears in Working Notes of AAAI Spring Symposium on Planning with Incomplete Information for Robot Problems Stanford, CA, March 1996, pp. 71-73

agent's control, while *action transitions* represent the intentional results of planned reactions. CIRCA can control the timing behavior of action transitions by adjusting the reaction timing constraints used by the Scheduler.

To build plans, CIRCA begins with a set of goal descriptions, a set of initial world states, and a set of transition descriptions that detail the types of events, actions, and processes possible in the world. The planning algorithm pushes the initial states onto a stack and then performs a modified STRIPS-like depth-first search for a plan that satisfies all the system's goals. On each planning loop iteration, the top state is popped off the stack and all applicable event and temporal transitions are applied, generating new reachable states that are pushed onto the stack. The planner uses a multi-step lookahead heuristic to choose the best action for the current state, generates the states that result from the selected action, and then repeats the planning loop. Chronological backtracking is initiated if the planner cannot find a good plan (e.g., if it cannot avoid a catastrophic failure state).

Uncertainty in Timing

Because CIRCA makes hard real time guarantees about its performance, and because the actual temporal extent of actions and exogenous processes cannot be known in advance, the world model is not intended to be a perfect representation of the world's actual behavior. Instead, CIRCA reasons about the world's *worst-case* timing behavior in order to build plans which are guaranteed to work in the worst case. For system-generated actions, the worst case is the *maximum* amount of time until the effects are realized, while for exogenous processes, the worst case is the *minimum* amount of time until a critical process status change can take place.

Because CIRCA only deals with a single worst-case timing value for each action and temporal transition, the process of manipulating this timing information is fairly simple. However, by carefully retaining enough information to plan preemptive reactions that deal with the domain's worst-case situations, this abstraction method still allows CIRCA to build reaction plans with guaranteed behavior.

Nondeterministic Transitions

Safety guarantees require that the space of possible states be completely described. However, making guarantees does not require any assessment of the probabilities or likelihoods of those possible states. One of the most common sources of uncertainty in robot planning problems is the tendency of robots to fail to successfully execute simple planned actions: wheels slip, sensors fail, grippers drop items, etc. Attempting to carefully characterize such failures can be very difficult, but proves unnecessary in mission-critical domains: if an action can fail, then its failure modes must be explicitly planned for. CIRCA represents this type of uncertainty using nondeterministic actions that implement a mapping from an input state to one of a set of possible output states, without incorporating probabilities. If such an action is planned, all the consequent states are generated and pushed onto the state stack, so that all possible outcomes must be planned for and made safe.

Thus nondeterministic actions are an extension of the worst-case abstraction used for timing information. Together, these worst-case assumptions form an extreme interpretation of Murphy's Law — "Anything bad that can happen will happen, at the worst possible time."

Uncertainty in Future Goals

Environmental uncertainty is a fundamental problem for any system. Overly optimistic assumptions about environmental predictability lead to plans that are quickly invalidated, while extreme pessimism disallows predictive planning altogether. Interleaving planning and execution so that sensory data can be collected as planning proceeds is one potential solution to this problem. This approach has been demonstrated in domains where uncertainty in initial conditions and sensor data precludes the immediate achievement of goals [1]. CIRCA can implement this method using feedback messages from the RTS to the AIS, passing sensor data acquired at runtime to the planner to affect the generation of future reactive plans.

Another type of problem arises when the uncertainty inherent in the environment dictates the actual set of *goals* that the system attempts to achieve. CIRCA uses its parallel AIS and RTS to manage such uncertainty in future goals. The AIS planner downloads reactive plans to the RTS to deal with a subset of the possible conditions, keeping the system safe while the planner reasons about the next set of possible conditions and necessary reactions. Changes in system goals can be managed by the planner while the RTS continues interacting with the world.

In order to ensure that the system remains safe and stable while the planner is searching for the next reactive plan, each plan is constructed to meet three objectives:

- Restrict the system to a given set of states.
- Ensure the system's safety in that set of states.
- Achieve the current goal(s).

The first two conditions ensure that once a set of planned reactions is being executed, the system is known to be safe for an indeterminate amount of time, during which the planner may generate the next set of planned reactions. Thus these conditions provide explicit "completeness" tests for plans that can be used, without loss of confidence, in architectures that plan and execute in an interleaved or parallel fashion.

Incomplete State Information

Reactive plan execution is desirable because the system responds to sensory data as opposed to an internal, potentially outdated model of the world. In the interests of efficiency and robustness, CIRCA's planner includes an unusual step that minimizes the precondition tests used by the planned reactions as much as possible, eliminating all sensory tests that are not absolutely required to disambiguate the states to which the various actions apply. As a result, the reactions executed by the RTS actually only test a subset of the total state features to determine whether they are applicable. In essence, the system has explicitly planned to acquire and deal with incomplete system state information.

Uncertainty in Initial Conditions

One of the easiest types of uncertainty for a reaction planner like CIRCA to handle is uncertainty in initial conditions. The system can handle any arbitrary set of initial states because they are just pushed onto the state stack at the start of planning process, and will thus be considered as reachable states that must be made safe by any feasible reactive plan. In other words, multiple possible initial states are treated the same as all other states that become reachable as the world model and plan are expanded.

During execution of the reactive plans, uncertainty in the system state is an issue for all states, initial or not. The reactive system must be able to disambiguate states in order to select the appropriate action. CIRCA's planner includes explicit consideration of the system's sensing capabilities, so that ambiguous states are recognized and avoided. CIRCA does not yet have approaches to dealing with worlds in which states cannot be accurately distinguished.

Recent Progress

Ongoing CIRCA research is investigating a number of system features and extensions. Work at the University of Michigan has included adding primitive probability information to transitions, allowing the system to make judgments about how useful certain sequences of planned actions are in resource-constrained situations. In addition, CIRCA has been interfaced to a flight simulator and has successfully demonstrated a variety of flight control operations including takeoff, point-to-point navigation, landing, and recovery from control action failures.

At Honeywell, CIRCA is being extended with domain-specific aircraft route planning capabilities, and, in cooperation with the University of Maryland, the CIRCA RTS is being ported to operate in true hard-real-time on the MARUTI operating system.

In addition, inherent difficulties associated with state-space explosion in the world model are being addressed. We have identified a general class of state-space *abstractions* having the special property that they preserve system safety, and are currently developing a framework in which a resource-limited planner like CIRCA can expand upon previously-abstracted details in a dynamic, context-sensitive manner. In this way, portions of the system's world model can be reasoned about or ignored depending on various contextual factors, but without compromising system safety.

References

[1] K. D. Krebsbach, *Rational Sensing for an AI Planner: A Cost-Based Approach*, PhD thesis, University of Minnesota, 1993.

[2] D. J. Musliner, E. H. Durfee, and K. G. Shin, "World Modeling for the Dynamic Construction of Real-Time Control Plans," *Artificial Intelligence*, vol. 74, no. 1, pp. 83–127, March 1995.