Northwestern University

The Institute for the Learning Sciences

TOWARD A QUALITATIVE THEORY OF SAFETY CONTROL - WHEN AND HOW TO PANIC INTELLIGENTLY

Technical Report # 14 • May, 1991

Dennis DeCoste

Established in 1989 with the support of The Arthur Andersen Worldwide Organization

TOWARD A QUALITATIVE THEORY OF SAFETY CONTROL - WHEN AND HOW TO PANIC INTELLIGENTLY

Dennis DeCoste

May, 1991

The Institute for the Learning Sciences
Northwestern University
Evanston, IL 60201

This research was supported by NASA Langley, under contract NASA-NAG-11023. The Institute for the Learning Sciences was established in 1989 with the support of Andersen Consulting, part of The Arthur Andersen Worldwide Organization. The Institute receives additional support from Ameritech, an Institute Partner, and from IBM.

Towards a Qualitative Theory of Safety Control — When and How to Panic Intelligently

Dennis DeCoste

Qualitative Reasoning Group
Institute for the Learning Sciences
Northwestern University
1890 Maple Avenue
Evanston, Illinois 60201
email: decoste@aristotle.ils.nwu.edu

Abstract

Safe control of a physical system requires the ability to both detect and avoid dangerous situations, while striving to achieve performance goals. State-of-the-art controllers still tend to rely heavily on classical control theory with feedback, occasionally with some limited use of associational-reasoning (expert systems) to help detect threatening situations and recall standard recovery procedures [Dvorak, 1987]. The need for model-based reasoning to control in novel situations and with incomplete data is widely acknowledged, but largely unaddressed to date. Existing relevant work has tended to focus on model-based monitoring to track the system state over time, with little attention to reasoning about control itself. In such work, any provisions for finding safe control actions tend to be mixtures of associational-reasoning and general-purpose planning or qualitative simulation — which suffer problems of brittleness and intractability, respectively.

To address those shortcomings, I propose a *qualitative theory of safety control.* The goal is to make explicit the kinds of intuitions that human operators use to focus the task of avoiding dangerous situations. I view the problem of safety control as two major issues: *when to worry* and *how to control.* Intuitively, one often worries about the need to control only if the chance of danger has increased — assuming the initial state was safe. Furthermore, different styles of control seem most appropriate in different types of situations. For example, when safety seems impossible to maintain it often seems best to try to at least delay the impending doom, in hopes that "buying time" might help. I argue that qualitative physics can be used to robustly capture such reasoning, but only with fundamental advances in how and what to qualitatively simulate.

1 Introduction

Especially in relatively well-engineered systems, such as airplanes and nuclear power plants, standard operating procedures are often sufficient for achieving one's performance objectives. For such systems, the fundamental responsibility of the human operator is *threat control* — detecting and evading novel threats to the expected achievement of the system performance goals. For example, a major reason for still having a human pilot in the cockpit these days is in case something goes wrong. In many cases, once a situation has stabilized (i.e. all critical threats are evaded), standard operating procedures are again sufficient. Yet, while technology continues to automate more of the routine achievement of goals, robust threat control remains elusive. This seems to be in part because even in highly-controlled domains, the ways that things could still go wrong are more subtle and plentiful than the ways to succeed.

1.1 Qualitative Physics

Robust reasoning about physical systems is a major motivation for the field of qualitative physics. Qualitative physics promises increased robustness in novel situations by using first principles of physics that offer wider coverage than the associational-rules of traditional expert systems. However, existing work in qualitative physics has had little to say about reasoning about threat control. The emphasis has been on developing representations and simulation techniques for explanation, steady-state diagnosis, and design.

In contrast to those more studied styles of qualitative reasoning, threat control poses some special difficulties:

1. *Threat detection involves anticipating faults before they happen* — whereas diagnosis has been viewed traditionally more as a task of explaining existing faults. Design also could involve anticipating faults, but existing work in qualitative physics has not addressed the issue of quality control in design.

2. *Threat control requires temporal projection under severe time constraints.* For explanation, diagnosis, and design tasks, it is often acceptable to incur the high cost of extensive qualitative simulation. However, it is usually critical in threat control tasks to quickly identify a potential threat and evaluate how it may progress over time — to provide an opportunity to evade the threat before it becomes reality.

I will argue that addressing these difficulties requires different sorts of qualitative simulation than have been provided by earlier work.

1.2 Planning

Reasoning about the dynamics of the physical world is only part of the problem of threat control — the other part is reasoning about control actions themselves. Traditionally, reasoning about actions is the domain of planning. Unfortunately, existing planning work is inadequate for threat control of physical systems.

1.2.1 Planning about Physical Systems

Part of the problem is that existing planners are not very good at physics. Much previous work in planning has either ignored modelling the dynamics of the physical world [Fikes & Nilsson, 1971] or employed means which suffered from poor modularity that precluded capturing many complex dynamic interactions [Hendrix, 1973].

Recent work has begun to explore the natural idea of of introducing qualitative physics into the planner. One such planner [Hogge, 1987] transforms qualitative physics models into rules and operators for a temporal planner like that of [Allen & Koomen, 1983]. Unfortunately, for the sake of tractability it seems necessary to make some simplifications which would be intolerable for threat control. For example, Hogge's rules assume that a positive influence will always result in a positive change, regardless of any other opposing influences.

Due to the general-purpose nature of such planners, accurate yet efficient reasoning about complex dynamic interactions seems difficult to achieve. However, such reasoning is exactly what qualitative simulation research is working towards. So, the converse alternative to Hogge's approach is to introduce actions into the qualitative simulation. Action-augmented envisioning [Forbus, 1989] does just that, hoping to capitalize on the efficient techniques developed for envisioning [Forbus, 1990]. Unfortunately, its focus on total envisioning is unrealistic to address the time-critical nature of threat control.

1.2.2 Planning With Negative Goals

Perhaps a more fundamental problem with using existing planners for threat control is that they are generally ill-suited for reasoning about negative goals. Research in domain-independent planning has tended to focus on achieving desired facts (goals), with little concern for the prevention of undesired facts (negative goals). A notable exception is Hogge's planner [Hogge, 1988]. In the case of action-augmented envisioning, reasoning is not goal-directed since it employs total envisioning.

Although negative goals may not be important in many planning domains, they are extremely useful for many threat control tasks. Negative goals seem to be a natural and tractable way to represent common threats to higher-level performance goals. For example, without an explicit negative goal to avoid explosions, keeping a boiler plant running would require inferring that explosions would clobber ways to achieve a smoothly running plant. Each negative goal indicates a negative condition to be avoided. Focusing on preventing negative conditions reduces the complexity

of threat control and avoids the need to even model the sometimes complex or even esoteric reasons why it is often best to avoid certain conditions.

1.2.3 Reactive Planning and Threat Control: A Good Partnership

To address issues of complexity and practicality, reactive planners [Agre & Chapman, 1987] [Schoppers, 1987] have recently become popular over more exhaustive classical planners [Chapman, 1985]. However, because of their increased shortsightedness, controllers using reactive planning to achieve their performance goals would have even greater need for threat control. An independent threat control mechanism could offer a sort of "safety net" to avoid novel disasters that a reactive controller might otherwise get itself into.

2 The Problem of Safety Control

What I am working towards is a means for providing just such a safety net. In this work, the focus is on a simplification of the threat control problem called *safety control*. In this view, one assumes that the system is initially in a *safe state*, one in which all its key performance goals are achievable. Safety control is the task of detecting potential ways that the system may be moving towards negative conditions and then proposing plans of control actions to prevent that from happening. The negative goals implicitly represent some of the threats to performance goals that a general threat controller would be trying to avoid. In particular, the negative goals represent those threats which would actually preclude achieving its key performance goals. In short, a safety controller acts to keep the system in safe states.

2.1 Negative Goals

Concentrating solely on preventing negative conditions promises to be more tractable than general threat control. This is because it amounts to only worrying about maintaining the achievability of the performance goals, without having to also show how they are achievable. This view is also consistent with the intuition that, in order to prevent unrecoverable errors, human operators tend to think more in terms of preventing "bad" states than ensuring "good" ones are maintained. This strategy seems appropriate for safety control because the point is to help avoid disasters which would prevent ever achieving the performance goals, not to maximize system performance.

Perhaps a more delicate issue is what should be the negative goals. Clearly, the answer is domain-specific — it depends on what behaviors need to be avoided. However, there are some general principles to consider. For one, context-dependency should be minimized by preferring higher-level negative goals for conditions that are universally bad. For example, using a negative goal to avoid explosions should be preferred over using a negative goal to avoid pressures high enough to cause some

explosions. Although human operators do sometimes simply reason about system parameters leaving nominal ranges, the intent here is to let the qualitative physics decide under which contexts potentially dangerous conditions, such as high pressure, should be avoided.

Thornier cases are conditions such as overflowing containers. It seems clear that a controller should "outlaw" explosions, but what about overflows? On the one hand, overflows are only really disastrous in cases such as the exported liquid doing damage to external devices or being unrecoverable and essential for system performance. But predicting whether such behavior will result from an overflow is often difficult or impossible. Obviously, the ideal answer involves weighing the chance versus the cost, but often such information is unavailable.

The tendency here is specify negative goals for conditions such as overflows, leaving it to a conflict resolver to override such goals only in those rare situations where avoiding such conditions would probably lead to even less desirable behavior. That would reduce the complexity of safety control for most cases yet still allow for novel solutions based on first principles for such rare ones. The good news for domains such as well-engineered systems is that such conflicts are probably much rarer than in everyday life, simply because designed artifacts are usually made to minimize the potential for such conflicts.

2.2 Separability of Safety Control and Quality Control

The underlying architecture for system control that I propose consists of two autonomous components: a safety controller and a *quality controller*. The safety controller would provide the safety net which prevents behavior that would unrecoverably preclude achievement of the key performance goals. This would free the quality controller, perhaps an opportunistic reactive planner, to concentrate on ways of best achieving the performance goals. The safety controller plays the role of a stabilizer — helping to bring the system to a functional state in response to novel dangers.

An underlying assumption of this architecture is that the key performance goals are still achievable as long as all negative goals are achieved. The intuition is that in many cases almost anything is possible as long as some fundamental constraints are not violated. For example, as long as a plane does not crash or lose its wings, it has a chance to land safely (a key performance goal). By being able to land safely, it can be refueled, flown to other destinations that were not optimally visited when it flew nearby earlier, and so on to finish satisfying the other performance goals.

Of course, things are not so simple in practice because of resource limitations. For example, making money is usually another key performance goal for an airline. So, it cannot typically afford to just land and refuel its planes whenever they run low on fuel due to poor quality control. One might be tempted to simply make avoiding non-positive net income a negative goal as well and let the safety controller worry about that condition ever becoming unavoidable. However, I do not expect this theory of safety control to be suitable for such problems. Unlike allowing safe

landings, a goal for non-positive net income is not related to the dynamics of the system. The proposed theory of safety control is intended to take advantage of the relatively high constraint in the dynamics of well-engineered systems. In general, it will probably not be of much use in preserving the achievability of goals when that depends more on the actions of external agents than the dynamics of the physical world. Coupling a theory of how external agents can impact the system with a theory of safety control could address such problems, but I will concentrate in this work on the safety control problem; it is a significantly hard problem all by itself.

Since many practical control problems do involve some such troublesome performance goals, the burden for making sure such goals can be achieved rests with the quality controller. It is therefore important that the safety controller minimizes its own control actions, allowing the quality controller maximum flexibility to control as required for optimal performance. Towards this end, my theory of safety control emphasizes controlling with patience, using qualitative simulation to carefully determine when further delay in control actions would conflict with the negative goals.

3 The Proposed Safety Control Framework

To address the above issues, I have begun to develop a framework for safety control, called SCOPS (safe control of physical systems). Figure 1 illustrates the basic structure of this framework. This framework suggests an incremental observe/interpret/worry/avoid cycle for processing observations over time. Each component is briefly highlighted by the following subsections.

3.1 The Simulator

The qualitative simulator provides the means for postdicting how new states could arise from previous states (interpretation) and predicting how current states could lead to unsafe states (worrying). It also provides the means for predicting the effects of control actions, as required by the avoider. In each case, simulation takes the role of incrementally building up the relevant portion of the implicit total envisionment for the system, referring to it to avoid resimulation whenever possible. As noted earlier, such simulation must be able to account for both the dynamics of physics and the effects of actions, especially control actions. I assume that a sound and complete model of the dynamics of the system is available and represented in terms of Qualitative Process Theory [Forbus, 1984].

3.2 The Monitor

Typically, the monitor provides the observations to the interpreter for processing. However, the monitor is also responsible for trying to obtain additional observations which can disambiguate alternative predicted behaviors, as requested by the worrier and the avoider.

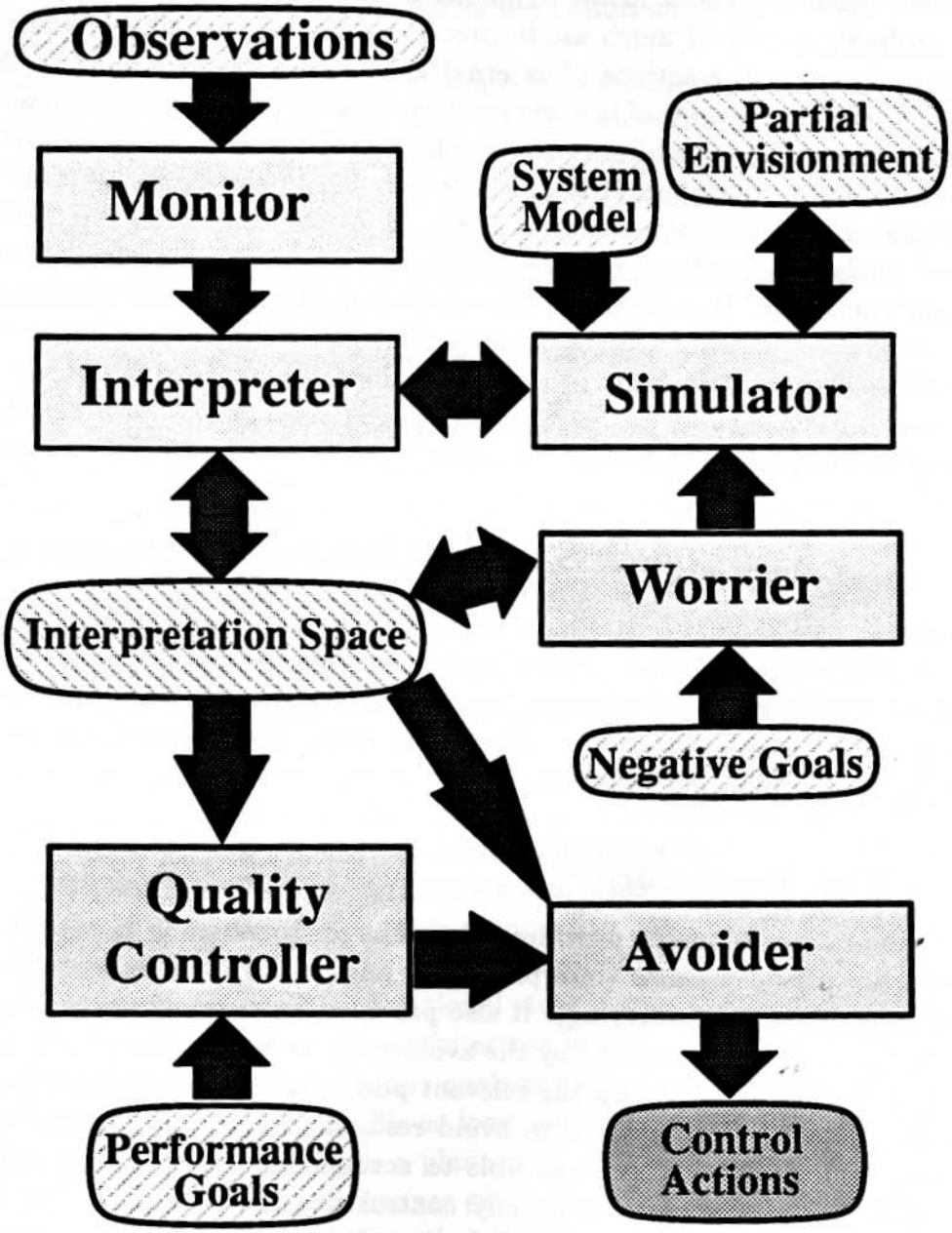

Figure 1: Overview of the SCOPS architecture

3.3 The Interpreter

Across-time observations are interpreted as possible states over time. The interpreter invokes the simulator to determine what kind of state the system might be in at the current time, based on the new observations and the previous states it might have been in. This is done using DATMI [DeCoste, 1989] [DeCoste, 1990], resulting in a space of alternative paths of states over time consistent with the incomplete observations and the ambiguous qualitative predictions of the simulator. However, for this task DATMI's simplification of using pre-compiled, total envisionments of total states to form this interpretation space is inappropriate. Consequently, this interpreter must maintain an interpretation space of partial states as the envisionment is incrementally generated by the simulator.

3.4 The Worrier

As system behavior is tracked by the interpreter, the worrier checks whether the system may have entered an unsafe state. Knowing when to worry about threats to safety first requires an operational definition of safety. The proposed theory of safety control suggests the following one:

Definition 3.1 (Safe state) *Let the states which contain the negative conditions for any negative goals be called the* negative states. *A state is considered* safe *when future negative states cannot arise due to the dynamics of the physical system alone.*

A state might then be shown to be safe by simulating to prove that no future behaviors lead to negative states.

Of course, determining safety by such exhaustive proof could be extremely expensive because simulation typically branches forward due to qualitative ambiguity and behaviors can span many states before reaching equilibrium or cycling back into previous states. In fact, this task is not even decidable unless the implicit total envisionment is finite.

To address this complexity problem, this framework relies on a key simplifying assumption: *the initial state of the system is safe.* This implies that any unsafe future state must arise due to some harmful external actions in safe states. Using this assumption, the worrier knows that control actions need only be considered when new dangers are indicated in the difference between the previous state (P) and the current one (C).

Determining whether the differences between P and C mean the system has entered an unsafe state can be broken into two steps. First, influence analysis on the differences are checked to see whether they could possibly influence the system more towards any of the negative states. Such analysis would basically consist of search through the graph of qualitative influences. If that search indicates C has no influences towards negative states that P did not also have, C can be considered safe as well.

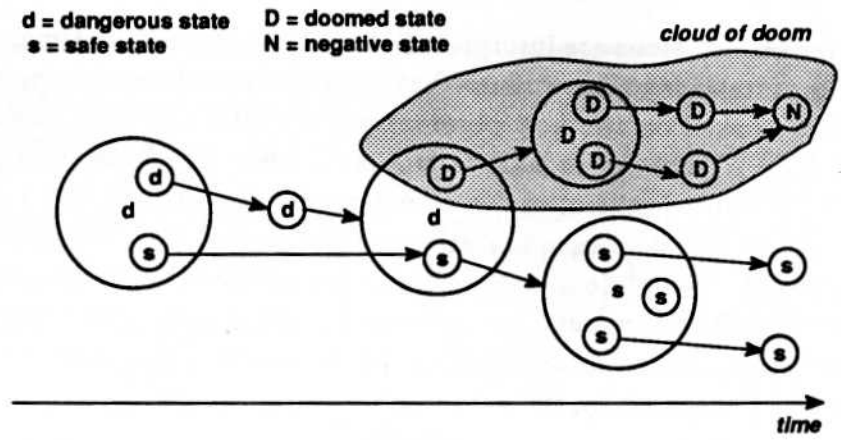

Figure 2: Example state history showing different types of states

Circles represent states and arrows represent transitions. Each circle is labelled at its center with the type of state it represents. Refinements of partial states are represented as circles within larger circles.

When influence analysis cannot determine that a state is safe, the worrier invokes qualitative simulation. Fortunately, the assumption that the initial state is safe can help reduce the complexity of this simulation as well. As each new state S is generated during simulation, the same type of influence analysis as suggested above can be used, with S instead of C now. One goal of this research is to assess when such influence analysis actually helps. Simulation continues until each path leads to a negative state or a safe state.

For further efficiency, a distinction can be made among unsafe states:

Definition 3.2 (Dangerous state) Dangerous states *have at least one way to reach a negative state that does not require any actions to occur. Furthermore,* doomed states *are dangerous states which cannot even avoid reaching negative states if control actions occur* [1]. *Negative states are themselves considered doomed. A state which is not dangerous is safe.*

This means that simulation down a path can also be terminated whenever a known doomed state is reached.

The result of all of this simulation is the determination of whether possible current state C is safe. Figure 2 provides a sample state history which illustrates how safe, dangerous, doomed, and negative states relate to one another.

In cases where the safety controller is thrown into the middle of a complex situation, the assumption that the initial state is safe might be quite disastrous. However,

[1] Control actions are considered to be just the subset of the external actions available to the safety controller. We do not prohibit the possibility of a heroic external agent saving us from an (apparently) doomed situation.

I suspect that in many cases it would not be unacceptable to start safety control when the system is at least likely to be in a safe state. If the initial state is not a doomed state, there is some hope for detecting danger before it becomes unavoidable.

When the assumption of initial safety seems inappropriate, one certainly could attempt to exhaustively prove (via qualitative simulation) that the initial state is safe. Such exhaustive proof seems especially undesirable for cases where simulation is so costly that heuristics for estimating safety are essential (see Section 5.1.3). In such cases, the assumption that the initial state is truely safe may be required for tractability.

3.5 The Avoider

When the worrier identifies a possible current state C as dangerous, the avoider is invoked to decide how to control to avoid following the paths from C which reach the negative states. If C is a doomed state, it might still be useful to try early control actions which may delay the occurrence of the negative state. The idea is to "buy time" in hopes that some external actions might fortuitously or intentionally be able return the system to some safe state if given enough extra time.

On the other hand, if dangerous state C is not doomed, it seems that it would often be best to postpone control actions as long as possible — that is, until just before a doomed state is reached. The intuition is that the actual behavior may fortuitously turn out to follow one of the paths leading to safe states. Due to ambiguity in qualitative simulation and incomplete observations, the particular behavioral path a system is following may not be known. Recall that in this framework the safety controller should strive to control only as necessary — to leave the quality controller as much room for optimizing as possible.

To decide which of the available control actions might be useful, the avoider can first invoke influence analysis to find candidate actions which either:

1. Defeat preconditions or quantity conditions of processes which are influencing the system towards the negative state. For example, turning off a pump which is pumping water into a container might help avoid the container overflowing.

2. Activate new processes which influence the system away from the negative state. For example, pumping water out of a container can help keep it from overflowing.

The simulator can then indicate which actions would return the system to a safe state. Since the interpreter will typically not have been able to uniquely identify the current state, the avoider should request additional data from the monitor to reduce the ambiguity. When ambiguity still remains, control actions which bring the most alternative states to safety should be preferred. Furthermore, alternative states which are "more dangerous", such as doomed ones or ones with more paths to negative states, should be given priority.

3.5.1 A Note About Doom

As doomed states are identified during simulations, a sort of "cloud of doom" will grow out from each negative state. In a sense, the safety controller is striving to avoid bumping into these clouds while allowing all other behaviors, to give the quality controller maximal space to work with. It might seem more efficient to just simulate backwards from the negative states and determine these clouds directly. Then, the expensive task of forward simulation to check if a state is safe could be avoided.

Although some amount of pre-growing of these clouds of doom might indeed prove very useful, I suspect that such backward simulation would not be preferable to the forward simulation advocated. Its condition for terminating simulation paths is identical to the forward simulation: when a safe or doomed state is reached. The problem with such backward simulation seems to be that there can be an enormous number of states which contain the negative condition of any negative goal — thus, there can be an enormous number of negative states from which to simulate backwards. Considering only negative states which are much like the current states might be a useful heuristic, but, then, so might be imposing depth bounds on forward simulation.

3.6 The Quality Controller

My theory of safety control has little to say about how the quality controller should work. The main concern is that the safety controller be able to deny an unsafe action suggested by the quality controller. This involves pretending that the action was performed and seeing if the current states would be dangerous. If dangerous states could result, the action might still be allowed if the avoider could find ways to bring the system back to safety.

4 Examples

The following three examples each demonstrate one of the three basic cases identified so far in this theory of safety control. These examples are all simple ones which our current qualitative theories of thermodynamics can model reasonably well. More real-world examples are also being developed, including some in the domain of aircraft engine failures [Schutte, 1990].

4.1 Example 1: Calmly Coming Face-to-Face With Doom, Part I

This example illustrates a case where a dangerous state is detected but control actions can be deferred until just before a doomed state would be reached. In this case the doomed state immediately precedes its negative state.

Scenario:

A pump-cycle consisting of two cans of water connected by a pump (from `CAN1` to

CAN2) and a valved pipe (see Figure 3).

Negative goals:

Avoid OVERFLOW(CAN1) being active.

Avoid OVERFLOW(CAN2) being active.

Available controls:

Turn the pump on or off.

Open or close the pipe valve.

Previous state:

WATER-LEVEL(CAN1) > WATER-LEVEL(CAN2).

Current observations:

WATER-LEVEL(CAN1) is dropping and WATER-LEVEL(CAN1) > WATER-LEVEL(CAN2).

Worrier:

Detects that there is more danger of OVERFLOW(CAN2) becoming active.

Possible current states:

The pump is on, or the valve is open, or both.

History trees from current states:

All paths lead to safe states — except for the state where the pump is on and the valve is closed, which can lead to the sole doomed state. (Actually, if HEIGHT(CAN2) < WATER-LEVEL(CAN1), then other states could also be dangerous.)

Doomed state:

CAN2 is full and WATER-LEVEL(CAN2) is rising.

Avoider:

Realizes that the system is not in a doomed state yet; discovers that the control action of turning off the pump, in the state S just before doomed state, would be sufficient to avoid reaching the doomed state.

Control action:

Turn off the pump, but only when the system enters state S (if ever).

Note:

If the amount of total water in boths cans is less than the capacity of CAN2, then there is actually no need to worry. This is a type of global constraint that would be nice to handle (see Section 5.2.4).

4.2 Example 2: Delaying Doom

This example illustrates the case where a dangerous state is detected which is doomed. In this case, the avoider suggests control actions to delay when the negative state is reached.

Scenario:

A pump-cycle consisting of two cans of water connected by a pump (from CAN1 to CAN2) and a valved pipe (see Figure 3).

Negative goals:

Avoid OVERFLOW(CAN1) being active.

Avoid OVERFLOW(CAN2) being active.

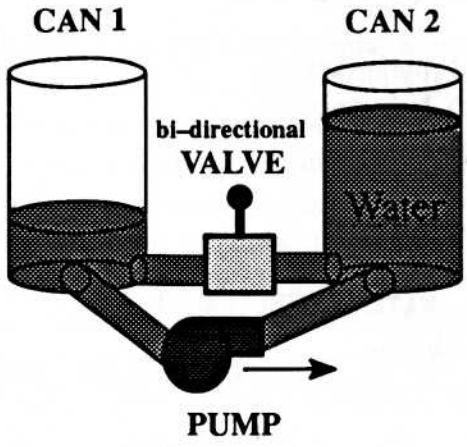

Figure 3: Two-can pump-cycle

Available controls:
Turn the pump on or off.
Open or close the pipe valve.

Previous state:
WATER-LEVEL(CAN1) is dropping and WATER-LEVEL(CAN1) > WATER-LEVEL(CAN2).

Current observations:
External water is flowing into CAN1 and WATER-LEVEL(CAN1) > WATER-LEVEL(CAN2).

Worrier:
Detects that there is more danger of either OVERFLOW(CAN1) or OVERFLOW(CAN2) becoming active.

Possible current states:
The pump is on, or the valve is open, or both. The relation between WATER-LEVEL(CAN1) and WATER-LEVEL(CAN2) can be anything, depending on the relative rates of the EXTERNAL-FLOW-INTO-CAN1, PIPE-FLOW, and PUMP-FLOW processes.

History trees from current states:
All paths lead to either OVERFLOW(CAN1) being active or OVERFLOW(CAN2) being active.

Doomed states:
All states in the history trees are doomed.

Avoider:
Realizes that system must now be in a doomed state; plans to delay doom, using reactions over time which depend on which state the system enters.

Control actions:
When WATER-LEVEL(CAN1) is rising: Turn on the pump. Open the valve unless WATER-LEVEL(CAN1) < WATER-LEVEL(CAN2). The idea is to minimize the net flow into CAN1, to maximize the time before OVERFLOW(CAN1) becomes active.
When WATER-LEVEL(CAN1) $\leq$ WATER-LEVEL(CAN2): Turn off the pump and open the

valve. The idea is to keep the water levels of the two cans equal, so that neither one overflows earlier than it has to.

4.3 Example 3: Calmly Coming Face-to-Face With Doom, Part II

This example again illustrates a case where a dangerous state is detected but control actions can be deferred until just before a doomed state is reached. However, unlike Example 1, this doomed state is not temporally adjacent to its negative state. Thus, this example demonstrates the importance of qualitative simulation in determining exactly how long control actions can safely be deferred.

Scenario:

A pump-cycle consisting of three cans of water. A valved pipe connects CAN1 with CAN2 and a pipe connecting CAN2 with CAN3 only allows water to flow through it from CAN3 to CAN2. The pump starts pumping water from CAN1 to CAN3 whenever WATER-LEVEL(CAN1) > TRIGGER-LEVEL. Once this pumping starts, it only stops when WATER-LEVEL(CAN1) = DESIRED-LEVEL. This pumping action is built into the system to help ensure that WATER-LEVEL(CAN1) stays below TRIGGER-LEVEL and near DESIRED-LEVEL.

Negative goals:

Avoid OVERFLOW(CAN1) being active.
Avoid OVERFLOW(CAN2) being active.
Avoid OVERFLOW(CAN3) being active.

Available controls:

Open or close the pipe valve.

The Problem:

If water flows from CAN2 to CAN1, then the controller might have to make sure to close the pipe valve before WATER-LEVEL(CAN1) > TRIGGER-LEVEL becomes true. Otherwise, the system might pump enough extra water into CAN2 to activate OVERFLOW(CAN2).

So, the worrier has to realize that if water is flowing from CAN2 to CAN1 and WATER-LEVEL(CAN3) $\geq$ WATER-LEVEL(CAN2), then there is subtle potential danger of OVERFLOW(CAN2) becoming active. The avoider cannot simply defer the control action (of closing the valved pipe) until CAN2 is just about ready to overflow — as was acceptable in Example 1 — because by that point the system might already be doomed. In particular, the pump could be relentlessly pumping water from CAN1 to CAN2, through CAN3, until WATER-LEVEL(CAN1) = DESIRED-LEVEL. Careful qualitative simulation, along with additional monitoring if possible, would indicate at what point control is required.

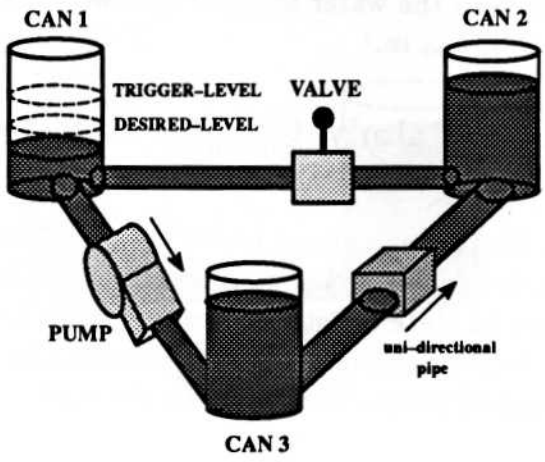

Figure 4: Three-can pump-cycle with a triggered pump

5 Summary of Major Advances Required

In this section, I discuss the major issues and potential approaches which I intend to explore during the development of this proposed theory of safety control.

5.1 Negative Goals and Safety

One fundamental issue to address concerns the utility of viewing safety control as the avoidance of negative conditions. The focus on negative goals in this proposed theory is due to an underlying subtle but significant distinction between "avoiding danger" and "maintaining safety". Negative goals represent dangerous conditions to avoid whereas their dual equivalents represent safe conditions to maintain. For example, if avoiding the overflow of a container is a negative goal, then the water level being below the top of the container would be the equivalent positive condition to maintain in all safe states. Although safe control could theoretically be realized with either view, the claim here is that avoiding negative conditions can often be more operational than maintaining positive conditions.

The intuition of this claim can be seen by contrasting the nature of clouds of doom with their dual: clouds of safety. Just as clouds of doom consist of all the doomed states, these clouds of safety consist of all the safe states. The proposed view of safe control is that of avoiding to enter the clouds of doom. The alternative view is that of staying within the clouds of safety. The intuition is that avoiding clouds of doom is more operational because the negative states provide suitable "seeds" for those clouds, whereas clouds of safety typically have no such seeds.

Some measure of the distance between the current state and a negative state can offer help in estimating whether a system is approaching a cloud of doom. To satisfy an analogous role, the seed of clouds of safety would have to represent the ultimate

safe states. That would allow estimates of whether the system is moving towards the fringes of the cloud of safety it is currently in. Unfortunately, such ultimate safe states are typically difficult to formulate or require numerous alternative states to specify adequately. For instance, consider the overflow example mentioned above. One ultimate safe state might involve the container being empty. However, some such states might actually be doomed; for example, the empty container might be under a running sink faucet which is stuck.

By allowing suitable distance-based estimates of whether things are getting worse, my view of safety control as the avoidance of negative conditions is especially useful for real-time safety control. In time-critical safety control tasks where the current state may be one of many alternatives (due to incomplete observations), it is perhaps far more important to quickly identify as many of the dangerous states as possible in the time allowed than classifying each state as safe or not. Using negative goals, search can be focused first on those simulation paths which have high estimates of leading to negative states. A dangerous state can then be identified as long as a single potential path to a doomed state is discovered.

5.1.1 Formulating Negative Goals

Despite the potential gains in focusing on negative goals, there are some serious open issues in how to best formulate them. One issue is that the set of appropriate negative goals may be somewhat contextually dependent, according to the mode of the system. During the taxi-ing stage, for example, avoiding the loss of cabin pressure might not be a negative goal to worry about, but during high-altitude cruising it might be appropriate to have such a negative goal.

Furthermore, in practice some negative goals may be more critical than others. For example, it might be best to avoid high turbulence during flight because such behavior presents significant risk of damage to the aircraft or to its stability. Since whether such damage will occur seems difficult to simulate with sufficient accuracy and such turbulence does not typically have any particularly redeeming qualities, high turbulence seems a reasonable condition to avoid using a negative goal. However, some situations may require accepting conditions such as high turbulence to avoid a higher-order negative condition, such as guaranteed damage to the wings. For example, one might have to avoid the wings hitting a mountain by subjecting the aircraft to a pocket of strong air currents above it.

Addressing these sorts of difficulties seems to require some sort of theory of the teleology of the negative goals. For example, this would require an understanding that high turbulence is a negative condition because it can easily lead to a higher-order negative condition such as aircraft damage. Then, such a negative condition might be tolerated only in attempts to avoid the higher-order dangers or when the risk of such higher-order dangers is counteracted by sufficiently high rewards.

An interesting possibility is to use extensive off-line envisioning as a decision tool in formulating appropriate negative goals. Envisioning could help determine the

clouds of doom for each candidate negative goal. An action-augmented envisionment of each particular mode of system behavior could be partitioned by these clouds of doom to see how well they isolate the normal system states from the undesirable states which fail to reach the key performance goals. This approach might be especially useful in domains where there is insufficient experience for suggesting an appropriate set of negative goals.

5.1.2 Safety versus Resiliency

A distinction should also be made between *safety*, as defined in this proposed theory of safety control, and *resiliency*, as defined below:

Definition 5.1 (Resilient state) *A state is considered* resilient *when future negative states cannot arise due to the dynamics of the physical system nor due to plausible actions or failures of system components.*

Thus, when the system is in a resilient state, it is not only safe but it is also unlikely to become unsafe even if actions or system faults occur.

The proposed SCOPS architecture does not attempt to keep the system in a resilient state because:

1. The simulation required for determining resiliency would be much more expensive than for determining safety. It would require that the worrier consider branches due to external actions and system faults while following potentially long simulation paths into the future.

2. In many cases, a sudden occurrence of a system fault or external action will not immediately move the system into a doomed state. The assumption is that there will be enough time after the danger first arises for the avoider to prevent a doomed state from occurring.

Nevertheless, it is important that the interpreter does consider how possible actions and system faults could lead to the current state. Without such considerations, the interpreter cannot provide the worrier with a comprehensive set of possible current sets. That could lead to the worrier not realizing that the system might be in a (unknown) dangerous state. Unlike the simulation required by the worrier, the simulation invoked by the interpreter to connect a current state with a state consistent with the previous observations will not typically involve long paths of states. Thus, branching for plausible faults and external actions should be much more likely to be tractable during interpretation.

There are still some types of branching not due to system dynamics which it might be both highly desirable and computationally acceptable for the worrier to consider. One type is control actions due to the normal operating procedures that would be used by the quality controller. When such procedures are represented as simple associational rules which map a state into one control action, little extra cost

would be incurred during simulation. In return for this small expense, this scheme would avoid worrying about dangers that the standard operating procedures can already handle.

Another type of branching which might be worth the cost to the worrier is those due to highly likely actions or faults which can easily lead to doomed states. For example, if one knows that a certain critical pump is acting up and its replacement will not be available for two days, it might be worth making sure that control actions are taken to avoid doom if the pump does happen to break. Although SCOPS is not intended to address the full problem of resiliency control, some small steps in that direction might be worth exploring in this work once the safety control foundation is more secure.

5.1.3 Estimating Safety

To accurately determine whether a state is safe or not, the worrier must exhaustively simulation from that state, to see if any doomed states will be reached. This can easily result in significant amounts of forward branching due to the ambiguity in qualitative simulation. Since such simulation is unlikely to be tractable in many real-world situations, it is of great practical importance to be able to estimate relative degrees of safety. By having already assumed that the initial state is safe, the worrier would then be willing to avoid costly simulation as long as the current state is estimated to be no more dangerous than its previous state.

Since many schemes for estimating safety are possible, one of the goals of this research is to organize the space of such schemes and explore their tradeoffs. Intuitively, a state's measure of safety might best be based on the relative level of expected danger, as follows:

Definition 5.2 (Relative safety) *Let the* expected danger *of a state be the sum, over each negative condition of the negative goals, of the product of the cost of that negative condition and the likelihood of the system degrading from that state into a negative state containing that negative condition. State A is considered* more dangerous *than state B if the expected danger of A is greater than the expected danger of B.*

In lieu of accurate numeric measures of such likelihoods and costs, these measures themselves must be estimated. The costs of negative conditions should be based on the teleology of the negative goals. For example, higher-order negative conditions should typically be more costly than lower-order ones.

The following intuitions suggest some heuristics for estimating the relative likelihood of a state being safe:

1. **Time and space metrics.** A state's chance of being safe is typically proportional to the distance between a state and a negative state. Thus, as a simulation path from a state increases in length without finding dangerous states, the

concern that the state is dangerous should generally decrease. This suggests that in time-critical tasks, where one must concentrate on finding the most dangerous states first, some sort of branch-and-bound simulation is probably best.

2. **Likelihoods of alternative branches.** Some branches due to qualitative ambiguity are more likely than others. Those branches which are more likely should contribute more to the measure of a state's safety. For example, a state should probably not be considered highly dangerous when the only way it can lead to a negative state is due to some coincidental behavior — such as two water levels becoming equal in two non-interacting subsystems and then those subsystems later interacting in such a way that those levels being equal matters.

3. **Percentage of doomed paths.** As the percentage of forward branching paths from a state that lead to doomed states increases, the chance of that state being safe tends to decrease.

Furthermore, when safety is being estimated instead of proven, the importance of resiliency increases. Therefore, it seems useful to have some means of estimating resiliency as well, such as:

1. **Degrees of freedom.** The expected resiliency of a state seems proportional to the number of control actions which would result in distinct states to which a state would otherwise not lead. The expected resiliency also seems to be proportional to how uniformly the control actions can scatter the future behavior within the space of states which are likely to be safe. The intuition is that when the system can be controlled to move from a state into almost any other state, the chance of the system being stuck in unsafe states is less.

2. **Time and space metrics.** The closer the danger is, the more likely it is that there will be no way to control the system to a safe state.

3. **Transformability.** If a sequence of control actions can be found for moving the system from the current state to a known safe state, then the current state can be considered safe as well.

5.2 Qualitative Simulation

Qualitative simulation plays a central role in SCOPS. Some of the more important and novel issues are mentioned in the following subsections.

5.2.1 Incremental Envisioning

Unlike current qualitative simulators [Kuipers, 1986] [Forbus, 1990] [Forbus, 1989], SCOPS's simulator must be able to incrementally generate the relevant portions of an implicit total envisionment as needed, without requiring the states to be totally

specified. Without such envisioning, the interpreter could not hope to efficiently maintain a concise interpretation space in most cases.

In a joint project, John Collins and I are developing just such an incremental envisioner, called IQE (Incremental Qualitative Envisioner) [DeCoste & Collins, 1991]. Abstractly, it is a hybrid of novel extensions to history generation [Kuipers, 1984], means-ends analysis, and action-augmented envisioning [Forbus, 1989]. In addition to our collaborations on this project, however, we are each developing independent extensions which are directed more towards our unique requirements. The following discussion of qualitative simulation solely concerns those unique requirements of my work on safety control.

5.2.2 Temporal Projection

Although IQE will be capable of some limited temporal inference, the SCOPS framework requires a richer capacity for temporal projection. For example, the notion of state transitions which each directly connect two envisionment states should be generalized to *achievable transitions* which can each represent the numerous alternative paths connecting two states. The motivation is that the interpreter and worrier do not always require knowing the exact path connecting two states, but simply whether some such path exists. The idea is to perform less detailed simulation when such distinctions are not required.

Another complex issue in temporal projection to address is how to inherit temporal constraints between other contexts to avoid the cost of recomputing them in similar contexts. Since the IQE framework is based on the use of an assumption-based truth maintenance system (ATMS) [de Kleer, 1986] [Collins & DeCoste, 1991], the ability to cache temporal constraints across contexts seems both desirable and possible. For example, if one ATMS environment E_1 is an immediate temporal successor of another environment E_2, then it might follow that subsuming environments of E_1 must immediately succeed subsuming environments of E_2.

5.2.3 Durations

In order for the avoider to use its strategy of delaying doom, durations of conditions and ways of influencing them must be represented. Two types of information seem required:

1. **Causal influences** — Determining which control actions can delay the occurrence of a condition, requires knowing the causal chains of influences on the duration of that condition. These influences might be determined in QP theory by reasoning about the influences on the rates of change in processes. The influences on the rates of change in processes in turn effect the durations of the processes themselves, which can then affect the durations of the process effects.

2. **Duration bounds** — Sometimes it is also important to know the particular range of time that some condition can hold. For one, this information can be useful to prune the interpretation space to remove ones inconsistent with duration constraints. Such bounds are similar to persistence times [McDermott, 1990], but the idea here is to also model the casual influences on those duration bounds in order to capture contextual constraints on persistences. I suspect that such modelling can be achieved by appropriately augmenting the language of process descriptions.

5.2.4 Global Constraints

Since not all global information is represented in the global states of envisionments, current qualitative simulators tend to suffer from occasional problems of global unsoundness [Kuipers, 1986]. This is of particular concern for safety control since the major objective is to only control when required. Globally unsound simulations can lead the safety controller to controlling in states that actually do not lead to doomed states.

For example, in Section 4.1 it was noted that an overflow need not really be worried about when the total amount of water in the two containers is less than the capacity of container CAN2. The simplest solution — comparing the capacity of CAN2 against the total amount of water in the entire system — would work in this case but would be be insufficient for more subtle cases. The general solution probably requires that the value for such an extra global variable be recorded in the particular states themselves as it changes over time. For example, if half of the water is removed and placed in some third container of the system, the relevant portion of the total water for determining whether the overflow of CAN2 is now only half even though the total water is still the same. The difficulty seems to be in deciding which extra global variables to store in the global states and when they are relevant. Without special care, the number of global states can exponentially explode to represent all possible combinations of values for all the extra global variables.

Another way that global constraints might be utilized is by noting whether some previously observed global cycles of behavior, such as an oscillation, could reflect a peak point in the possible behavior. For example, if one saw a ball bounce up and down in a room without hitting the ceiling, one would not expect it to do so on successive bounces. One would no longer worry about the ball hitting the ceiling until something happens which could causally influence this situation — such as the room being rocketed away from the earth's gravity or the ceiling being pushed down. Although the answer seems to be to somehow represent the total energy of the ball as part of the global state, such extra global variables must be used with extreme care, as noted above.

The use of extra global variables to address this problem is not new. In fact, Kuipers has argued for the use of distinct values for these extra global variables, called landmarks [Kuipers, 1986], to extend the quantity space as potentially relevant

new values are detected during history generation. However, the unresolved issue to address here is how to tame the introduction of such landmarks to reduce the danger of combinatoric explosions mentioned earlier.

5.3 Teleology of controls

To most effectively utilize the available controls, the safety controller should have some knowledge of the teleology of those controls. Knowing why a control was designed into the system is one very strong clue for when to use it. Furthermore, when several control actions seem promising, there is intuitive appeal to preferring the one which applies in fewer contexts than the other control actions. The idea is that that must be one of the few cases where that control action is most appropriate. In the case of well-engineered systems for which one can assume the designer was competent and had a purpose for each control, this view may have considerable merit.

Although the actual purpose and nature of each control is largely domain-specific, several key dimensions of controls seem especially useful to reason about:

1. **utility context** — defined by the performance limitations of the control. For example, pumps usually have limits to how high of a pressure difference they can maintain.

2. **operating context** — defined by how the control action is performed. For example, closing a valve may be considerably more difficult when the water current is strong. Also, if the operator is too far away from a valve and a control action is required immediately, a closer control must be sought — perhaps even to just delay the danger to buy time for reaching the valve.

3. **reliability** — whether the control will be in working order when needed. In contexts where a control is suspected to be unreliable and alternative controls are not suitable, strategies such as waiting until just before doom must become more conservative to provide a margin of error.

4. **start-up time** — the time between when a control is invoked and when it begins to function. Obviously, one should also not wait until just before doom to invoke a control that will take some time to start-up.

In all of these cases, the hope is that by modelling the controls as part of the system as a whole, these dimensions can be reasoned about in a robust way.

6 Summary of Major Assumptions and Claims

1. Negative goals can adequately define safety.

2. The system starts in a safe state.

3. A sound and complete model of the system is available.

4. Safety control and quality control can be performed largely independently.

7 Related Work

This section highlights some of the related work which has not already been discussed above.

7.1 Interpretation

SCOPS's interpreter builds on both my earlier DATMI work [DeCoste, 1989] and the MIMIC process monitoring work [Dvorak & Kuipers, 1989]. Both systems address the important problem of tracking the system state over time by dynamically pruning the space of interpretations as observations are made. DATMI employs a top-down approach, mapping the system envisionment onto the observed history to find the full set of possible current states, even when the observations are very incomplete. In contrast, MIMIC uses a more bottom-up approach, using the observations to drive the incremental simulation as it tries to casually relate previous observations with the current ones. In order to take advantage of the full potential of the incremental envisioning used by SCOPS's simulator, the interpreter combines DATMI's use of envisionments and MIMIC's use of incremental history generation.

7.2 Qualitative Physics

Despite some work in using teleological knowledge of system components to guide explanation [deKleer, 1984], diagnosis, and design [Franke, 1989] of physical systems, little attention seems to have been given to how teleological knowledge of controls can be used along with causal models of those controls in control tasks.

Reasoning about how an external action or system fault may move a system more toward a negative condition is a form of *comparative analysis* [Weld, 1988]. Whereas qualitative simulation is the prediction of behavior over time, comparative analysis is the assessment of how an earlier perturbation would affect that behavior. For example, comparative analysis could determine that using a heavy block in a spring-block system would increase the period of oscillation.

However, Weld's differential qualitative (DQ) analysis and exaggeration techniques for comparative analysis are insufficient for SCOPS's worrier. As Weld points out, there are many cases where his techniques cannot determine the effect of a perturbation even though qualitative simulation of the perturbed system would. For example, DQ cannot determine that an increase in the pumping rate for the pump-cycle of Figure 3 would increase the water level of CAN1 at equilibrium. Yet, in order to avoid an overflow, a safety controller must be able to make just such a conclusion.

Although these techniques are insufficient for a theory of when to worry in safety control, they could still play a useful role. In particular, they could at least let the worrier avoid expensive qualitative simulation in those situations when such weak techniques are still sufficient for determining that observed perturbations cause changes away from the negative conditions.

7.3 Planning

The notion that safety control is the task of maintaining safety (see Section 5.1) suggests a relation to Gervasio's work on *achievability proofs* [Gervasio, 1990]. These achievability proofs are able to assert that a goal is achievable without actually having to determine a specific plan of actions to do so. The issue is whether analogous *maintainability proofs* could be formulated to determine whether a state is safe without requiring simulation. For example, if a key performance goal is having food to eat, being near a grocery store might ensure safety with respect to that goal. This seems analogous to Gervasio's *multiple opportunities* class of achievability proofs.

However, as Gervasio admits, such proofs are perhaps better viewed as merely plausible ones since they are typically defeasible in practice. Her call for reasoning about degrees of achievability may be partially addressed by heuristics for estimating safety, such as those proposed in Section 5.1.3.

In most earlier work in planning, the need for practical temporal projection was largely ignored. Such projection is especially important for tasks such as safety control where the exact current state is usually not known with much certainty. Hanks' recent work [Hanks, 1990] argues for the utility of *bundling* states when they differ only in ways which are irrelevant to the objective of the projection. The hope is that through careful use of such bundling, exponential explosions due to forward branching in the projection tree can be avoided. This bundling technique seems to be similar to the use of partial states, as opposed to totally-specified ones, in our incremental envisioner `IQE`.

The work of Dean and Boddy also shares many of the underlying motivations of my `SCOPS` work. Like our `IQE` work, they are working towards incremental causal reasoning [Dean & Boddy, 1987] which is both tractable and avoids the weak, often unsound, projections commonly made by most nonlinear planners. Their analysis of *anytime* algorithms [Dean & Boddy, 1988] presents a useful framework for exploring problems such as time-critical safety control.

Wellman's notion of *tradeoff formulation* based on qualitative probabilistic networks [Wellman, 1990] is relevant to both `SCOPS`'s worrier and avoider. For real-world problems, safety must typically be estimated, either because the observations are uncertain or because they are too incomplete to allow tractable simulation of the many alternative consistent behaviors to prove safety. Accurately estimating safety in such cases requires resolving tradeoffs, such as whether to view an observation as reliable or not and whether to reduce ambiguity in the observations by assuming some unobserved condition holds. As Wellman emphasizes, resolving such tradeoffs is often best

handled by first identifying what the real tradeoffs are. For instance, when trying to avoid an overflow of CAN2 for the system of Figure 3, the worrier does not require knowledge of the water level of CAN2 when the pump is off and the valve is closed.

Probabilistic notions of tradeoff formulation, such as Wellman's, are certainly necessary for a general solution to optimal control. However, a weaker, non-probabilistic notion may often be sufficient for safe control, especially of well-engineered systems. For example, even if there is some danger that an aircraft engine may fail during flight, it is not necessarily more dangerous to proceed towards a take-off. It might be *sub-optimal* to proceed all the way down the runway when the danger of the engine failing during take-off is very likely. But it might nevertheless be *safe* to do so, as long as take-off can be safely aborted if the danger still exists right before take-off. In SCOPS, being in a dangerous state is acceptable – as long as there will be a state before any future doomed states in which evasive action can be taken to return to a safe state. Even in Wellman's domain of medical therapy, such procrastination is sometimes possible. For example, people often risk poor health by depriving themselves of some sleep or food, realizing that any ill-effects can be detected and treated if necessary.

Nevertheless, SCOPS's non-probabilistic formulation of tradeoffs can result in overly-conservative safety control – such as aborting a take-off even when the danger of the engine failing during the take-off is very low or the likelihood of being able to land safely if it did fail is very high. Whereas Wellman's probabilistic tradeoff formulations identify conditions under which a plan must be replaced with another because it is *inadmissible*, analogous formulations for SCOPS could be useful to identify conditions under which a plan of control would be *unnecessary*.

8 Acknowledgements

Thanks to Ken Forbus and John Collins for useful comments. This research has been supported by NASA Langley, under contract NASA-NAG-11023.

References

[Agre & Chapman, 1987] Agre, P and Chapman, D. Pengi: an implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 268–272, July 1987.

[Allen & Koomen, 1983] Allen, J. F and Koomen, J. A. Planning using a temporal world model. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 741–747, August 1983.

[Chapman, 1985] Chapman, D. *Planning for Conjunctive Goals.* Master's thesis, Massachusetts Institute of Technology, November 1985. (AI Technical Report 802).

[Collins & DeCoste, 1991] Collins, J and DeCoste, D. CATMS: an ATMS which avoids label explosions. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, July 1991. To appear.

[Dean & Boddy, 1987] Dean, T and Boddy, M. Incremental causal reasoning. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 196–201, July 1987.

[Dean & Boddy, 1988] Dean, T and Boddy, M. An analysis of time-dependent planning. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 49–54, August 1988.

[DeCoste & Collins, 1991] DeCoste, D and Collins, J. IQE: an incremental qualitative envisioner. In *Proceedings of the Fifth Workshop on Qualitative Physics*, May 1991. To appear.

[DeCoste, 1989] DeCoste, D. *Dynamic Across-Time Measurement Interpretation: Maintaining Qualitative Understandings of Physical System Behavior.* Master's thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, October 1989. (Technical Report UIUCDCS-R-90-1572, University of Illinois at Urbana-Champaign, February 1990).

[DeCoste, 1990] DeCoste, D. Dynamic across-time measurement interpretation. In *Proceedings of the Nineth National Conference on Artificial Intelligence*, pages 373–379, July 1990.

[deKleer, 1984] deKleer, J. How circuits work. In D Bobrow (Ed.), *Qualitative Reasoning about Physical Systems*, The MIT Press, 1984.

[de Kleer, 1986] de Kleer, J. An assumption-based TMS. *Artificial Intelligence*, **28**(2), March 1986.

[Dvorak & Kuipers, 1989] Dvorak, D and Kuipers, B. Model-based monitoring of dynamic systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1238–1243, August 1989.

[Dvorak, 1987] Dvorak, D. *Expert Systems for Monitoring and Control.* Technical Report AI 87-55, Artificial Intelligence Lab, University of Texas at Austin, May 1987.

[Fikes & Nilsson, 1971] Fikes, R and Nilsson, N. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, **2**:189–208, 1971.

[Forbus, 1984] Forbus, K. D. Qualitative process theory. *Artificial Intelligence*, **24**:85–168, 1984.

[Forbus, 1989] Forbus, K. D. Introducing actions into qualitative simulation. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1273–1278, August 1989. (Technical Report UIUCDCS-R-88-1452, University of Illinois at Urbana-Champaign, August 1988).

[Forbus, 1990] Forbus, K. D. The qualitative process engine. In D. S Weld and J de Kleer (Eds.), *Readings in Qualitative Reasoning about Physical Systems*, pages 220–235, Morgan Kaufmann, 1990. (Technical Report UIUCDCS-R-86-1288, University of Illinois at Urbana-Champaign, December 1986).

[Franke, 1989] Franke, D. W. Representing and aquiring telological descriptions. In *Proceedings of the 1989 Workshop on Model Based Reasoning*, pages 62–67, August 1989.

[Gervasio, 1990] Gervasio, M. *Learning Completable Reactive Plans Through Achievability Proofs.* Master's thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, May 1990. (Technical Report UIUCDCS-R-90-1605, University of Illinois at Urbana-Champaign, May 1990).

[Hanks, 1990] Hanks, S. Practical temporal projection. In *Proceedings of the Nineth National Conference on Artificial Intelligence*, pages 158–163, August 1990.

[Hendrix, 1973] Hendrix, G. Modelling simultaneous actions and continuous processes. *Artificial Intelligence*, 4:145–180, 1973.

[Hogge, 1987] Hogge, J. Compiling plan operators from domains expressed in qualitative process theory. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, July 1987.

[Hogge, 1988] Hogge, J. Prevention techniques for a temporal planner. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 43–48, August 1988.

[Kuipers, 1984] Kuipers, B. Commonsense reasoning about causality: Deriving behavior from structure. In D Bobrow (Ed.), *Qualitative Reasoning about Physical Systems*, The MIT Press, 1984.

[Kuipers, 1986] Kuipers, B. Qualitative simulation. *Artificial Intelligence*, 29:289–338, 1986.

[McDermott, 1990] McDermott, D. A temporal logic for reasoning about process and plans. In J Allen, J Hendler, and A Tate (Eds.), *Readings in Planning*, pages 436–463, Morgan Kaufmann, 1990.

[Schoppers, 1987] Schoppers, M. Universal plans for reactive robots in unpredictable environments. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 1039–1046, August 1987.

[Schutte, 1990] Schutte, P. Unpublished NASA Langley document. 1990. Summarizes the actual data and expert explanations of eleven commercial aircraft accidents due to a variety of engine failures.

[Weld, 1988] Weld, D. *Theories of Comparative Analysis.* PhD thesis, Massachusetts Institute of Technology, May 1988.

[Wellman, 1990] Wellman, M. P. *Formulation of Tradeoffs in Planning under Uncertainty.* Pittman and Morgan Kaufmann, 1990.