Iteration against Locally-Determined States

for Qualitative Monitoring and Diagnosis

Ethan Scarl

Advanced Technology Center Boeing Computer Services Bellevue, Washington

Abstract

The proliferation of possible qualitative states can be drastically reduced in some simulation contexts by incorporating information from observation. When internal states of the model cannot be directly assigned from system inputs or other boundary conditions, the distinction between monitoring and diagnosis can become blurred, merging into the common problem of finding a consistent version of the simulation model. The number of possible states is reduced by the constraints introduced by sensor measurements and may be further reduced by serial iteration in the case where component states are locally determined but neither controlled nor measured directly. This iteration should usually terminate very quickly, especially when a good selection of initial (default) states is known, and leads to a relatively efficient simulation. It may be expedient to separate out this class of constraints for checking the results of simulation, as opposed to incorporating it into the running of the qualitative model.

Introduction

Qualitative Reasoning often proceeds by discretizing parameters and behaviors that are inherently continuous. However, some components or subsystems lend themselves to qualitative reasoning by having a natural set of discrete states which are more suitable for characterizing behavior than the range of continuous variation with those states.

In the *projective* simulations often studied (e.g., Kuipers 86; Forbus 85), the objective is to *predict* the evolution of system state from a known initial configuration through plausible successor states, without intervention or additional information becoming available during the simulation. Such systems typically suffer from a proliferation of possible states as the uncertainties in the occurrence, outcome, or timing of critical landmark changepoints compound each other. Of course, this can be reduced by reconvergent branching, intra-state constraints, state transition constraints, on up to global system constraints, but the residual number often remains necessarily large since the qualitative model does not contain the information needed to eliminate all unrealistic behaviors (Kuipers 86).

Applied simulations are usually intended to compare their simulated behaviors with real physical systems, however, either to monitor for faulty physical behavior or to test the adequacy of the model. Here too, the objective is to select a vector of component qualitative states that together describe the system. These need to be consistent with each other and with system boundary conditions (static simulation), or a path through the space of such vectors (dynamic simulation), so that all parameters in the vector change state in a globally consistent way. Here too, the branching factor can be high due to local uncertainty, ambiguity, or intrinsic inability to distinguish between predicted outcomes. The control and reduction of such explosive numbers has been the object of considerable effort within the QR community (e.g., Kuipers and Chiu 87).

It is the objective of this paper to propose methods which will produce correct and reasonably chosen conclusions while avoiding combinatoric search within a restricted range of circumstances. We naturally do not claim to achieve completeness without exhaustive search, but show how a single coherent or satisficing model can be achieved for an important class of objects: those whose qualitative state is determined entirely by the parameters at its connections with the rest of the system.

The next section discusses the way that observation can be used to monitor with ambiguous models and the role of diagnosis. The following sections show how the use of state/parameter constraints can reduce combinatorics in simulating systems with many such objects. The paper concludes with an example from a realistic flow system.

Observations as Constraints

The use of observation to constrain or modify a simulation model, common in model-based diagnosis, can also reduce the number of possible expected behaviors in qualitative simulation. For example, the usual qualitative simulation of a ball that is thrown up in the air will predict that it either keeps rising or stops. When it actually stops at time t', any possibility of its rising after t' can be rejected. An *observation constraint*, requiring a parameter to have its observed qualitative value at a particular time, is propagated along with the system's predefined constraints by whatever inference engine drives the simulation. As in all local constraint propagation, the addition of more constraints should cause solution sets to be smaller and found more quickly.

Of course, if the observation prunes *all* qualitative possibilities, this is a mismatch between observation and simulation that requires diagnosis to resolve. At the expense of completeness, the most common and economical monitoring assumption is that diagnosis is unnecessary as long as *any* qualitative possibilities remain consistent with measurements.

Using observation to constrain simulation is the basic premise of model-based diagnosis (Davis and Hamscher 88; De Kleer and Williams 87; Scarl 87). Sometimes predefined alternative fault models explicitly replace part of the simulation (e.g., Kuipers 87, de Kleer and Williams 89), but model-based diagnosis frequently predefines only expected behaviors and uses observational evidence as a guide to model modification. However, integrating observation with simulation is less common as a pruning technique for monitoring than in diagnosis, perhaps because it may seem to compromise the independence of the simulation. Some workers have used measurements to distinguish between alternative models (Forbus 87; Kuipers 87), but at this point they are doing diagnosis rather than monitoring. We might wish to carry specific types of failure modes along, however, effectively moving certain considerations from diagnosis back into monitoring. We could explicitly represent the possibility that an observation is itself in error, for example. Such possibilities can simply propagate as additional hypotheses or branch points in the simulation, as they do in diagnosis (Scarl 89).

In the following sections, confirmation by observation is used to markedly reduce the proliferation of behavior for a useful class of components.

Components with Locally Determined State

The complexity of simulation can be reduced for a class of system components whose states *cannot* be directly determined by commands or observations. This is particularly appropriate to the "diagnostic monitoring" problem discussed in the previous section, where monitoring potentially produces many possible states, and the objective is to avoid having to generate all possible sensor predictions before pruning with data from observation. It *is* diagnostic in that it modifies the model to bring it into agreement with observation. It is *not* diagnostic in that it neither assumes nor derives any fault models or incorrect behaviors. The "diagnosis," if you will, is limited to selecting the qualitative state among the component's properly working states in which it is now operating. Rather than generating all possibilities, one may begin by simply assuming an operating state and then "diagnosing" that assumption if it incorrectly predicts observable data.

Let us use the term *Components with Locally-determined State* (CLSs) to mean components that have the following characteristics:

- Component behaviors are classifiable into a well-defined set of qualitative states {Qi}, selected so that:
- Component state is determined, in correct operation, by the vector **x** of the component's inputs, outputs, or other local parameters that constrain or are constrained by the its behavior. That is, there is some function $f(\mathbf{x})$ which uniquely determines the component state Qi.

CLSs may or may not have an associated *default state*, that is expected to determine its properly operating behavior in the current environment. Component "malfunctions" are limited to their not being in their assumed states.

We will refer to

 $Qi = f(\mathbf{x})$

as the Parameter-State Relation (PSR) between some particular state Qi of a CLS and its local parameters. Stated as a relation, this is a test of consistency between state and parameters. Used as a function, it produces the state consistent with x.

The following qualitative iteration can now be performed to select appropriate states for CLSs:

Assume that any CLSs with default states are now in those states. For each remaining CLS, select a state arbitrarily from its set of possibilities. All components now have assigned a single qualitative state, and therefore a single simulation behavior, regardless of how many possible values may be envisioned as qualitative alternatives or quantitative ranges for its local parameters. The model is now sufficiently constrained by

1. boundary conditions or system inputs,

2. observations, and

3. state assumptions for the CLSs

so that, for each assumption of parameters being propagated to the CLSs, the values of remaining parameters can be computed, and any derived system state can be tested for consistency.

Parameter-State Relations and a Satisficing Method

The computation and consistency checks just described can have different outcomes:

If no consistent state of the model as a whole can be found: This is the most difficult, in the sense that the identification of CLSs yields minimal help. Standard diagnostic methods (e.g., Davis 85; Davis and Hamscher 88: De Kleer and Williams 87; Scarl 87) can be invoked to resolve the discrepancy. So long as an incomplete set of solutions is acceptable, so that not all consistent diagnoses or simulation models need be determined, suspending or simply reassigning the more arbitrary CLS state assignments is a reasonable heuristic to begin the search for a consistent solution. We are, at least, no worse off than if we had enumerated all possible states to begin with. If no CLS state reassignment yields a consistent solution, then a failure has been detected, and the usual (expensive) diagnostic candidate generation techniques must be invoked. The PSRs have not yet been used explicitly since predictions do not yet exist for the local parameters \mathbf{x} . Once a consistent configuration of the model as a whole has been reached, they can be readily checked.

If the PSR constraints for the CLS are all satisfied: the current state assignment for the CLSs is compatible with observation. The usual cost/completeness tradeoff will de-

termine whether alternative state configurations should also be sought. A compromise strategy is to try different state assignments for CLSs whose states were assigned arbitrarily or whose state determination has been marked as critical by designers, while retaining the hypothesis that defaulted non-critical CLSs are working properly.

If the PSRs of some of the CLSs are violated: use the PSR as a function to assign to each of these CLSs the state indicated by its newly determined local parameters. This is where the major payoff of using CLSs is found. Rerun the model with these state assignments, as a second iteration.

If in this second iteration no consistent model can be found: a failure has been detected. We have good reason to believe that some or all of those CLSs which changed state between iterations are malfunctioning, since a consistent model was found "only" with these in wrong states. Naturally, it is possible that there is a totally different set of faults which could also explain both iteration behaviors; we claim only to have located one plausible fault candidate set.

If in this second iteration the model is again consistent, with the PSR constraints for all CLSs now being satisfied: another possible set of state assignments has been found. No conclusion can be reached about the health of the CLSs, however, since both unexpected (first iteration) and expected (second iteration) CLS configurations agree with observation. The changed CLSs may be faulty, but the effect of their faults has been masked.

If the PSRs of any CLSs remain unsatisfied: the unsatisfied PSR constraints must again be examined to see which state changes they prescribe for the CLSs. This leads to another iteration run, unless it would lead back a previously tested model configuration; in the latter case we would have stumbled into a pathologically cyclic and presumably rare behavior. The outcome of such further iteration would be interpreted in the same way as the second iteration. Such a sequence appears unlikely to produce very many iterations without achieving full PSR satisfaction, an inconsistent model, or a cycle-back to a previous state.

Note that this method is applicable whether the underlying simulation is qualitative or quantitative, but the CLS states *must* be qualitatively distinguished.

An Example

Figure 1 shows 2 pumps (marked "SIP") driving fluid through an array of pipes and valves. This is typical of part of the emergency water supply system found in some nuclear power plants. The details are unimportant here, but those who like to count will find in the diagram 21 Check Valves (CVs; 15 are shown and 6 more implied at the center of the right edge) and 4 Pressure-Operated Relief Valves (PORVs, marked "PRT" in the figure). These are the CLSs. Each CV has two states, *open* or *closed*, depending on the sign of the pressure difference across it. PORVs are usually considered to have three states: *closed* if the pressure difference is significantly less than a threshold **Po**, *open* if the pressure difference is significantly greater than **Po**, or *regulating* with its pressure difference within a

Pressure-Operated Relief Valve (PORV)

+ + + Check Valve (CV)



Figure 1. Part of a Pump and Valve System

small interval around **Po**. Depending upon which states they are in, the CVs and PORVs contribute different constraints upon system operation; these cannot be consolidated easily into single descriptions except by the use of expressions conditional upon the pressure drop. Such conditionals are not readily handled by flow solvers, so we need to solve the system model in some particular configuration of CV and PORV states. With 21 CVs and 4 PORVs, however, there are $2^{21}*3^4$ or almost 170 million possible state configurations.

Now, the PORVs have very definite default conditions: they are used here for emergency protection rather than pressure regulation, and so they are expected to be closed. Most of the CVs default open, except those at the lower right which only open if the downstream pressure drops substantially. It happens that all the CLSs have clear default positions, although some of these may change if the environment (e.g., the downstream backpressure) changes significantly.

By the method described, the system solution will be found using sensor measurements, valve and other commanded constraints, the pump operating characteristics, and boundary conditions like the upstream and downstream pressures. Even if there are PORV or CV failures, this solution will be largely validated on a single iteration. Suppose that the PSR constraints for all CVs and PORVs are satisfied: the pressure drop is positive across CVs assigned open, negative across those assigned closed, and that the drop across all PORVs is significantly less than **Po**. If so, we are finished. Of course, if any of these would not affect a sensor reading even if they were in a different state, that possibility will go unnoticed.

On the other hand, suppose that two of the PORVs have gone into their regulating state unexpectedly and measurements *have been* affected. Driving the model with those measurements should lead to PSR constraint failure at the PORVs, with pressure drops near or greater than **Po**. It is quite possible that only one of their pressure drops will show up at this point as being within tolerance of **Po**. Rerunning the model with that PORV's state as regulating would expose the pressure difference violation at the other PORV.

Three iterations is a considerable improvement over 170 million. If the third iteration fails to arrive at a consistent state of the model, then a viable conclusion is that the second PORV has not satisfied its PSR constraint and is in a faulty condition. There are the caveats, discussed above, about masking, pathological conditions, or other fault conditions not related to CV and PORV states. Should these occur and a full analysis be practical, at least that analysis can proceed with relatively little effort having been lost on state-determination for the CVs and PORVs.

Conclusion

When we can use observed measurements of an evolving physical system to constrain its simulation, the number of qualitative possibilities may be significantly reduced. When we know what has actually happened, we may no longer need to carry information about all the things that didn't happen. In particular, observations can be combined with knowledge of how local parameters determine the qualitative state of components. This information can be used to correct wrong initial state assumptions for these components, with a quick convergence to a coherent model, without having to carry all possibilities through the simulation. We are currently investigating the extension of this work to include objects whose qualitative description is determined by internal state parameters, in addition to the local parameters at its ports.

REFERENCES

- (Davis 85) R. Davis, "Diagnostic Reasoning Based on Structure and Behavior," in D. G. Bobrow (ed.), Qualitative Reasoning about Physical Systems, MIT Press, Cambridge, MA, 1985.
- (Davis and Hamscher 88) R. Davis and W. Hamscher, "Model-based Reasoning: Troubleshooting," in H. E. Shrobe (ed.), Exploring Artificial Intelligence: Survey Talks from the National Conferences on Artificial Intelligence, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.
- (de Kleer and Williams 87) J. de Kleer and B. C. Williams, "Diagnosing Multiple Faults," Artificial Intelligence, 32(1), 1987.
- (de Kleer and Williams 89) J. de Kleer and B. C. Williams, "Diagnosis with Behavioral Modes," in Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89), Detroit, MI, August, 1989.
- (Forbus 85) K. Forbus, "Qualitative Process Theory," in D. G. Bobrow (ed.), Qualitative Reasoning about Physical Systems, MIT Press, Cambridge, MA, 1985.
- (Forbus 87) K. Forbus, "Interpreting Observations of Physical Systems," *IEEE Transactions on System, Man, and Cybernetics*, SMC-17, May/June 1987.
- (Kuipers 86) B. Kuipers, "Qualitative Simulation," Artificial Intelligence, 29, 1986.
- (Kuipers 87) B. Kuipers, "Qualitative Simulation as Causal Explanation," *IEEE Transactions on System, Man, and Cybernetics*, SMC-17, May/June 1987.
- (Kuipers and Chiu 87) B. Kuipers and C. Chiu, "Taming Intractible Branching in Qualitative Simulation," Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87), Milan, August, 1987.
- (Scarl 87) E. A. Scarl, J. R. Jamieson, and C. I. Delaune, "Diagnosis and Sensor Validation through Knowledge of Structure and Function," *IEEE Transactions* on System, Man, and Cybernetics, SMC-17, May/June 1987.
- (Scarl 89) E. A. Scarl, "Sensor Failure and Missing Data," Proceedings of the first AAAI-sponsored Workshop in Model-Based Reasoning, E. A. Scarl, ed., at the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89), Detroit, MI, August, 1989. Alternatively, E. A. Scarl, "Diagnostic Tolerance for Missing Sensor Data," Telematics and Informatics, 6 (3/4), November/December, 1989.